

박사학위논문

설명 가능한 인공지능 기반의  
프로세스 분석 자동화 연구

2021년

한 성 대 학 교 대 학 원

스마트융합컨설팅학과

스마트융합제품전공

정 찬 일



박사학위논문  
지도교수 이후진

# 설명 가능한 인공지능 기반의 프로세스 분석 자동화 연구

Study of Process Analysis Automation  
based on XAI

2021년 6월 일

한성대학교 대학원

스마트융합건설링학과

스마트융합제품전공

정 찬 일

박사학위논문  
지도교수 이후진

# 설명 가능한 인공지능 기반의 프로세스 분석 자동화 연구

Study of Process Analysis Automation  
based on XAI

위 논문을 공학 박사학위논문으로 제출함

2021년 6월 일

한 성 대 학 교 대 학 원

스마트융합컨설팅학과

스마트융합제품전공

정 찬 일

정찬일의 공학 박사학위논문을 인준함

2021년 6월 일

심사위원장 \_\_\_\_\_(인)

심사위원 \_\_\_\_\_(인)

심사위원 \_\_\_\_\_(인)

심사위원 \_\_\_\_\_(인)

심사위원 \_\_\_\_\_(인)

# 국 문 초 록

## 설명 가능한 인공지능 기반의 프로세스 분석 자동화 연구

한 성 대 학 교 대 학 원  
스 마 트 융 합 컨 설 텅 학 과  
스 마 트 융 합 제 품 전 공  
정 찬 일

최근 인공지능의 발전과 함께 예전에는 사람만이 할 수 있다고 생각되어지던 이미지 인식, 고장 예측 등의 분야들이 자동화되고 있다. 그러나 아직 많은 업무 영역에서는 사람의 개입이 꼭 필요하다고 여겨지고 있으며, 경영혁신을 위한 방법론 중 하나인 업무 프로세스 혁신 (Process Innovation) 프로젝트는 전문가의 판단과정 없이는 수행이 불가능하다고 여겨지던 영역 중 하나이다. 업무 프로세스 혁신은 현황분석을 통한 이슈 도출과 그 이슈의 원인을 분석하여 목표 프로세스 모델을 수립하는 단계로 나뉜다. 그중 프로세스 마이닝은 정보시스템 로그를 이용하여 현행 프로세스 모델을 자동으로 생성해줌으로써 현황분석을 효과적으로 수행하는 데 크게 이바지했다. 그러나 이슈에 대한 근본 원인이 무엇인지 파악하기 위해서는 여전히 전문가의 경험과 직관이 매우 중요하다고 인식되고 있다.

본 연구에서는 프로세스 마이닝의 데이터를 기반으로 업무 프로세스 수행 결과가 정상일지, 또는 이슈가 발생할지를 머신러닝 모델을 이용해

사전에 예측하고, 학습된 예측 모델을 다시 설명 가능한 인공지능 알고리즘을 사용하여 분석함으로써 이슈의 원인을 자동으로 찾아냈다. 이렇게 업무 프로세스 이슈의 원인을 자동으로 도출함으로써 기존에는 사람의 개입 없이 불가능하다고 여겨지던 업무혁신을 위한 이슈 원인분석 영역까지 인공지능의 활용 범위를 확장하였을 뿐만 아니라, 기업들이 많은 시간과 비용을 들여 진행하던 업무 프로세스 혁신 프로젝트를 훨씬 적은 인력과 시간만으로도 더 정확하게 수행할 수 있게 되어 기업의 지속적인 혁신에 이바지할 수 있을 것이다.

연구는 은행, 전자제품 제조 기업, 인터넷 서비스 기업에서 공개한 프로세스 마이닝의 수행 결과와 시스템 로그 데이터를 대상으로 실험을 진행하였다. 먼저 은행의 여신 프로세스를 기반으로 고객의 대출 신청 취소가 발생하는 원인을 분석하여 알고리즘을 통한 원인분석 자동화가 가능하다는 것을 증명하였다. 또한, 제조 기업의 환불 프로세스를 대상으로 딥러닝을 포함한 다양한 머신러닝 알고리즘과 아직은 연구가 활발히 진행 중인 설명 가능한 인공지능 분야의 여러 알고리즘을 적용해본 후 비교 분석하였으며, 전통적인 머신러닝 모델보다는 딥러닝 모델을 사용할 때 예측의 정확도는 높아지지만, 반대로 원인분석에는 전통적인 머신러닝 모델이 유리한 Trade-off 관계가 있음을 확인하였다. 그러나 결과에 영향을 미치는 가장 중요한 요인은 데이터의 전처리로, 프로세스 로그를 그대로 사용하지 않고 프로세스 흐름, 수행 시간, 추가적인 속성들을 포함할수록 원인분석 알고리즘의 정확도가 높아짐을 확인하였다. 이 결과를 기반으로 인터넷 서비스 기업의 모바일 앱 사용자 로그를 기반으로 프로세스 모델을 생성한 후 머신러닝 모델을 적용해 사용자가 이용권 결제에 실패하는 경우를 미리 예측하고 설명 가능한 인공지능 알고리즘을 통해 결제에 실패하게 되는 원인을 자동으로 도출하였다. 자동으로 도출된 실패의 원인은 기존에 전문가들이 참여하여 분석한 실패의 원인과 동일하였으며, 이를 통해 이슈 원인분석 자동화 알고리즘이 전문가의 분석을 대체할 만큼 우수한 성능을 갖고 있음을 확인하였다.

본 연구를 통해 제시된 ‘설명 가능한 인공지능 기반의 프로세스 분석 자동화’는 그동안 이미지 인식, 결과 예측 등에 주로 활용되던 인공지능

알고리즘의 활용 범위를 사람의 판단과정 없이는 불가능하다고 생각했던 이슈 원인분석 분야까지 확장하였다. 또한, 현행 프로세스 분석 중심의 프로세스 마이닝 도구를 원인분석까지 자동화하도록 개선함으로써 유용성을 강화하였다. 이를 통해 업종 간 경쟁이 치열해지고 점점 빠르게 변화하고 있는 현대 경영환경에서 기업의 생존을 위해 필수적인 프로세스 혁신 활동을 훨씬 적은 시간과 인력으로 수행할 수 있고, 나아가 자동화를 통한 상시적인 프로세스 혁신 체계까지 가능하게 함으로써 기업의 경쟁력 강화와 지속적인 성장에 실질적인 도움이 될 것이다.

향후 ‘설명 가능한 인공지능 기반의 프로세스 이슈 원인분석 자동화’ 기법을 다양한 업무 프로세스 데이터에 적용하여 신뢰성을 높이는 연구를 진행하면서, 데이터 전처리와 적용 알고리즘을 표준화하여 프로세스 마이닝 솔루션의 주요한 기능으로 추가될 수 있도록 개발을 진행할 것이다. 더불어 프로세스 마이닝뿐 아니라 원인분석이 필요한 다양한 업무에 설명 가능한 인공지능 기반의 원인분석 기법을 적용해본다면 다양한 학술적, 경제적 효과를 얻을 수 있을 것이다.

**【주제어】** 설명 가능한 인공지능, 프로세스 마이닝, 원인분석 자동화

# 목 차

제 1 장 서론 .....	1
제 1 절 연구의 배경 및 필요성 .....	1
제 2 절 연구의 목적 .....	5
제 2 절 연구 범위 및 논문 구성 .....	7
1) 연구의 범위 .....	7
2) 논문의 구성 .....	8
제 2 장 선행연구 및 한계 .....	11
제 1 절 경영 혁신과 프로세스 마이닝 .....	11
1) 현대 경영과 프로세스 혁신 .....	11
2) 프로세스 마이닝을 통한 프로세스 모델 생성 .....	14
제 2 절 프로세스 마이닝과 머신러닝 .....	29
1) 머신러닝 알고리즘과 프로세스 .....	29
2) 머신러닝 기반의 프로세스 예측 .....	32
제 3 절 설명 가능한 인공지능 .....	33
1) 설명 가능한 인공지능의 개념 및 유형 .....	33
2) LIME (Local Interpretable Model-agnostic Explanations) .....	35
3) SHAP (SHapley Additive exPlanations) .....	37
4) SA (Sensitivity Analysis)와 LRP (Layer-wise Relevance Propagation) .....	39
제 3 장 연구방법 및 절차 .....	42
제 1 절 가설 및 연구 방법 .....	42
제 2 절 연구 데이터 수집 및 변환 .....	45
제 3 절 연구 도구 및 라이브러리 .....	49

제 4 장 실험 및 분석 결과 .....	53
제 1 절 은행 여신 프로세스 분석 자동화 .....	53
1) 금융 데이터 특성 및 전문가 분석 결과 .....	53
2) 프로세스 데이터 전처리 .....	60
3) 프로세스 결과 예측 .....	64
4) 프로세스 원인분석 .....	68
5) 기존 연구 결과와 비교 .....	70
5) 실험 결과 및 의미 .....	74
제 2 절 제조 기업의 환불 프로세스 분석 .....	75
1) 대상 프로세스 특성 및 전문가 분석 결과 .....	75
2) 프로세스 데이터 전처리 .....	82
3) 딥러닝 기반 예측 및 XAI 알고리즘 적용 .....	85
4) 머신러닝 기반 예측 및 XAI 알고리즘 적용 .....	93
5) 프로세스 흐름 기반 영향도 분석 .....	96
6) 실험 결과 및 의미 .....	97
제 3 절 모바일앱 사용 프로세스 분석 .....	98
1) 모바일 앱 프로세스 특성 및 전문가 분석 결과 .....	98
2) 데이터 전처리 .....	100
3) 예측 모델 작성 .....	106
4) 이슈 원인분석 결과 .....	107
5) 실험 결과 및 의미 .....	110
제 5 장 결론 .....	111
제 1 절 연구 요약 .....	111
제 2 절 연구의 기여 및 시사점 .....	112
제 3 절 연구의 한계 및 향후 계획 .....	114
참 고 문 헌 .....	116
ABSTRACT .....	124

## 표 목 차

[표 2-1] 프로세스 마이닝 구성 요소 .....	15
[표 2-2] 이벤트 로그 필수 요소 .....	16
[표 2-3] 프로세스 분석 기법 .....	20
[표 2-4] 프로세스 분석 방법론 .....	27
[표 2-5] 기존 방법론과의 차이 .....	29
[표 2-6] 알고리즘별 프로세스 예측 정확도 .....	33
[표 3-1] 프로세스 분석 예시 .....	43
[표 3-2] 사용 데이터 세트 및 포맷 .....	47
[표 4-1] 사용 데이터 세트 특성 .....	53
[표 4-2] 이벤트 타입 설명 .....	55
[표 4-3] 고객 취소 프로세스 통계 .....	60
[표 4-4] Activity 인코딩용 Dictionary .....	61
[표 4-5] 마지막 Activity 유형 .....	63
[표 4-6] Activity 인코딩 결과 예시 .....	63
[표 4-7] Hyper Parameter .....	65
[표 4-8] 예측 모델 정확도 평가 .....	67
[표 4-9] BPIC에서 사용된 분석 기법 .....	71
[표 4-10] BPIC에서 사용된 알고리즘 .....	72
[표 4-11] 환불 프로세스 데이터 세트 특성 .....	75
[표 4-12] 환불 프로세스 구성 Activity .....	76
[표 4-13] 전문가가 분석한 주요 이슈 .....	79
[표 4-14] 상위 10개 Variant .....	83
[표 4-15] Activity 수치화 (Word Index) .....	84

[표 4-16] LSTM 모델 Activity별 평균 영향도 .....	89
[표 4-17] 예측 모델별 예측 정확도 비교 .....	93
[표 4-18] Activity흐름 기반 영향도 분석 .....	96
[표 4-19] 모바일 앱 사용자 로그 데이터 세트 특성 .....	101
[표 4-20] 모바일 앱 Activity 특성 .....	102
[표 4-21] 모바일 앱 버전 데이터 특성 .....	104
[표 4-22] 모바일 앱 로그 데이터 전처리 결과 .....	105
[표 4-23] 결제 실패 프로세스의 원인분석 결과 .....	108

## 그림 목 차

[그림 1-1] PI 방법론의 혁신 필요성 .....	2
[그림 1-2] 프로세스 원인분석 자동화 발전 단계 .....	4
[그림 1-3] 프로세스 혁신 자동화 단계 .....	6
[그림 1-4] 프로세스 예측과 원인분석의 차이 .....	7
[그림 2-1] PI 프로젝트 수행 방법론 예시 .....	13
[그림 2-2] PI에서 프로세스 마이닝 적용 .....	17
[그림 2-3] 프로세스 모델 자동 생성 .....	18
[그림 2-4] 프로세스 마이닝 모델의 복잡성 .....	19
[그림 2-5] 프로세스 마이닝 분석 예시 .....	21
[그림 2-6] 조직 분석 예시 .....	22
[그림 2-7] 프로세스 성능 분석 예시 .....	23
[그림 2-8] KPI와 PPI의 관계 .....	24
[그림 2-9] 프로세스 마이닝 분석 방법 .....	25
[그림 2-10] KPMG의 프로세스 마이닝 수행 방법론 .....	26
[그림 2-11] 프로세스 마이닝 기법을 활용한 모니터링 및 시각화 구현을 위한 프레임워크의 설계 .....	28
[그림 2-12] RNN의 장기의존성 문제 .....	31
[그림 2-13] LSTM의 구조 .....	32
[그림 2-14] Cell State의 구조 .....	32
[그림 2-15] XAI 알고리즘 분류 .....	35
[그림 2-16] 글로벌해석력과 로컬해석력 .....	36
[그림 2-17] SHAP 알고리즘 작동방식 .....	38
[그림 2-18] LRP 알고리즘 작동방식 .....	40
[그림 3-1] 연구 가설 .....	42
[그림 3-2] 증명 방법 및 절차 .....	44

[그림 3-3] 데이터 전처리 사례 (Theis, & Darabi, 2019) .....	48
[그림 3-4] 구글 Colab 환경 연결 .....	49
[그림 3-5] PM4PY 라이브러리 설치 .....	50
[그림 3-6] XES 포맷 데이터 읽기 .....	50
[그림 3-7] 머신러닝 라이브러리 사용 .....	51
[그림 3-8] 예측 모델 생성 .....	51
[그림 3-9] XAI 라이브러리 설치 .....	52
[그림 4-1] 상위 30개 Variant .....	54
[그림 4-2] Activity 목록 및 실행 건수 .....	56
[그림 4-3] 주요 프로세스 흐름 .....	57
[그림 4-4] 신청건 대비 성공 비율 .....	57
[그림 4-5] 프로세스 마이닝 모델 (DISCO 활용) .....	58
[그림 4-6] 예측 모델의 구조 요약 .....	66
[그림 4-7] 예측 모델 학습 곡선 .....	67
[그림 4-8] 고객 취소 원인분석 결과 #1 .....	68
[그림 4-9] 고객 취소 원인분석 결과 #2 .....	69
[그림 4-10] 환불 프로세스 마이닝 모델 (DISCO로 작성) .....	77
[그림 4-11] 요약된 환불 프로세스 모델 .....	78
[그림 4-12] 콜센터와 인터넷 환불 프로세스 비교 .....	80
[그림 4-13] Performance Analysis 분석 .....	81
[그림 4-14] Case 별 처리 시간 그래프 .....	82
[그림 4-15] Case 별 Activity 개수 .....	83
[그림 4-16] 실제 프로세스 Case 및 인코딩 결과 .....	85
[그림 4-17] LSTM 예측 모델의 파라미터 .....	86
[그림 4-18] LIME 알고리즘 기반 프로세스 분석 #1 .....	87
[그림 4-19] LIME 알고리즘 기반 프로세스 분석 #2 .....	88
[그림 4-20] SHAP 알고리즘 적용 결과 (Force Plot) .....	90
[그림 4-21] SHAP 알고리즘 적용 결과 .....	90

[그림 4-22] SA와 LRP 알고리즘 적용 결과 .....	92
[그림 4-23] Decision Tree모델의 가중치 .....	94
[그림 4-24] LIME 기반 해석 결과 .....	95
[그림 4-25] SHAP 기반 해석 결과 .....	95
[그림 4-26] 이슈 원인분석 성능에 대한 영향도 .....	98
[그림 4-27] 음원 서비스 결제 프로세스 .....	100
[그림 4-28] 예측 모델 구성 파라미터 .....	107
[그림 4-29] 요소별 영향도 .....	109
[그림 4-30] 전문가와 알고리즘 분석 결과 비교 .....	110

# 제 1 장 서론

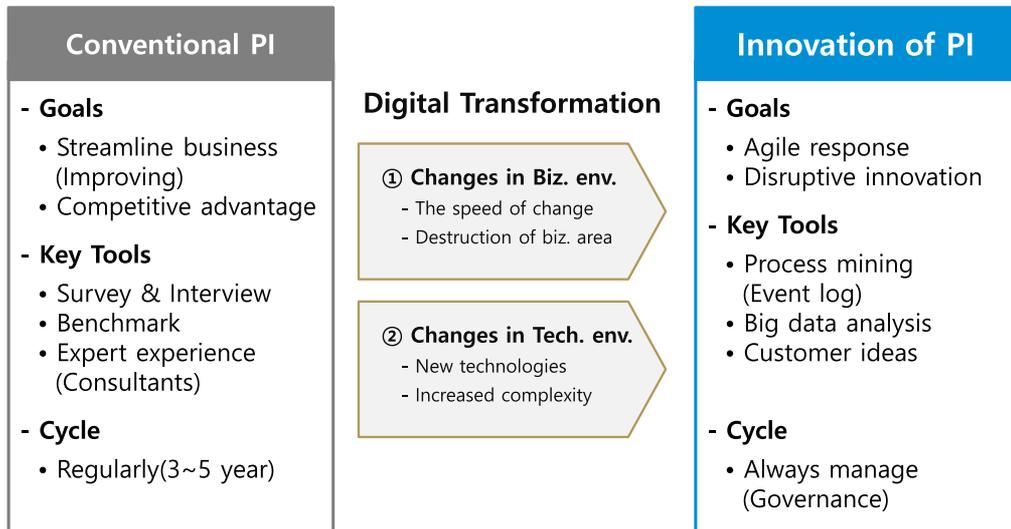
## 제 1 절 연구의 배경 및 필요성

기업 업무에서 컴퓨터 사용이 일상화되고, 인터넷, 모바일을 통한 소셜 네트워크가 발전하면서 인공지능 학습에 필요한 빅데이터를 쉽게 확보할 수 있게 되었고 대량의 데이터를 병렬로 처리할 수 있는 GPU와 컴퓨팅 파워가 향상되면서 인공지능에 관한 연구가 급속히 발전하고 있다. 예전에는 사람만이 할 수 있다고 생각되던 이미지 인식, 자연어 처리, 예측 등의 분야에서 인공지능은 이미 인간을 뛰어넘는 성능을 보여주고 있다. 그러나 아직 많은 영역에서 사람의 판단과 개입이 꼭 필요하다고 여겨지고 있으며, 경영혁신을 위한 방법론 중 하나인 업무 프로세스 혁신 (Process Innovation) 프로젝트는 전문가의 경험과 역량에 의한 직관적인 판단 없이는 수행할 수 없다고 여겨지는 업무 분야 중 하나이다.

끊임없이 변화하는 경영환경에서 업무혁신을 통한 경쟁력 확보는 기업에 있어 생존을 위한 필수적인 조건이다. 특히 업무혁신을 통한 기업 프로세스의 최적화는 매우 중요한 요소이다. 이를 위해 많은 경영 이론에서는 업무혁신을 위한 프로세스 및 방법론을 제시하였다. 특히 Hammer (1990)가 제시한 BPR (Business Process Reengineering)이나 Davenport (1993)의 PI (Process Innovation) 방법론은 경영혁신을 위한 방법으로 현재의 업무 프로세스 (As-Is Business Process)를 모델링하고 업무혁신을 위한 목표 업무 프로세스 (To-Be Business Process)를 설계하도록 제시하였으며, 현재에도 많은 기업이 PI(Process Innovation) 방법론에 따라 업무혁신 프로젝트를 진행하고 있다.

그러나 디지털 탈바꿈 (Digital Transformation)이라는 현대 경영환경의 변화 속에서 기존의 경쟁우위를 위한 업무 효율화라는 PI 프로젝트의 목적이 이제는 기민한 대응과 파괴적 혁신이라는 목적으로 변화하고 있다. 경영환경의 변화가 빨라지고 업종 간의 경계가 사라지는 시장 상황과 인공지능과 빅데이터, 클라우드를 비롯한 다양한 신기술의 등장 및 기존 정보시스템의 복잡

도가 증가하는 기술환경의 변화는 PI 프로젝트의 변화를 요구하고 있다. [그림 1-1]처럼 설문, 인터뷰, 벤치마킹과 전문가의 경험에 의존하던 기존의 PI에서 다양한 정량적인 분석 기술을 적극적으로 활용하고, 외부 인력이 아닌 내부 직원과 고객의 의견을 중심으로, 주기적으로 큰 비용과 인력을 들여 프로젝트를 진행하는 것이 아닌 거버넌스 관점에서 지속해서 관리할 수 있는 체계로의 변화가 필요한 시점이다.



[그림 1-1] PI 방법론의 혁신 필요성  
[Fig. 1-1] Innovation of PI Methodology

특히 업무혁신을 위한 대부분의 방법론에서 제시하는 도구들은 수행 절차와 인터뷰 기법, 체크 리스트와 같이 정성적인 수준에 머물러 있어 프로젝트 수행자의 경험과 개인적 역량에 의한 직관에 깊이 의존하고 있다. 다시 말하면 같은 업무혁신 방법론에 따라 동일한 절차로 프로젝트를 진행하더라도 참여하는 사람들의 경험과 역량에 따라 전혀 다른 결과가 도출될 수도 있다는 점이 가장 큰 문제로 지적된다.

이러한 문제점을 해결하고자 Van Der Aalst (2011)는 정보시스템에서 생성된 로그를 수집, 분석하여 업무 프로세스를 자동으로 모델링하고 업무 프로세스상의 문제를 정량적으로 분석하는 프로세스 마이닝 기법을 제시하였다.

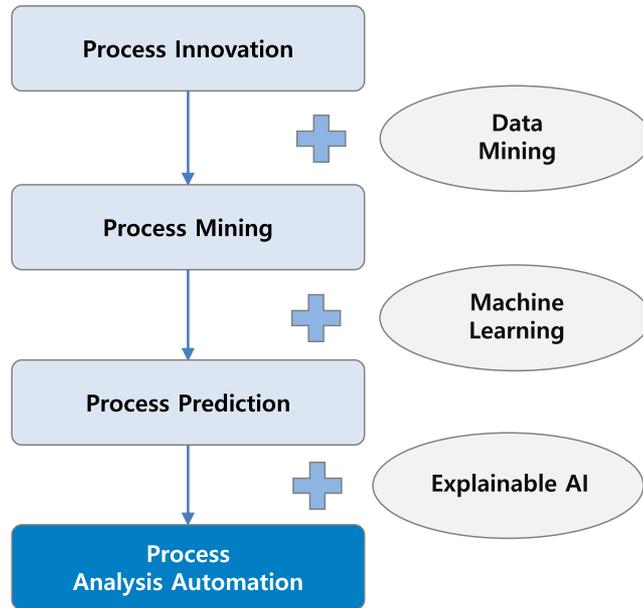
정보시스템의 활용이 점점 늘고 있는 업무환경의 변화와 빅데이터 기반의 다양한 분석 기술 발전 등으로 프로세스 마이닝은 점점 더 주목받고 있으며 실제로 많은 기업에서 도입하여 성과를 얻고 있다. 그러나 프로세스 마이닝에서 업무 시스템 로그로부터 프로세스 모델을 자동으로 생성 (Process Discovery) 하는 데에는 다양한 알고리즘과 도구가 개발되어 있으며 많은 부분이 자동화되었지만, 생성된 프로세스 모델을 분석하여 이슈가 발생하는 원인을 찾아내고 개선방안을 도출하는 프로세스 개선 (Enhancement) 영역에서는 아직도 전문가의 직관에 의존하고 있다.

프로세스 마이닝 도구의 분석 기능 역시 전문가의 판단을 지원하기 위한 시각화 기능 위주로 구성되어 있다. 시스템 로그를 사용해서 도출해 낸 프로세스 모델은 기본적으로 매우 복잡한 거미줄 형태로 표현되므로, 프로세스 마이닝 도구는 이것을 사람이 이해 가능한 형태로 시각화하기 위해 프로세스를 추상화할 수 있는 다양한 필터 기능을 제공하고 있다. 그러나 이런 추상화 과정에서 데이터의 왜곡이 발생하거나 정작 중요한 비정상 프로세스가 제외되기도 한다. 무엇보다 프로세스 마이닝 도구를 사용하더라도 이슈의 원인을 찾아내는 과정은 결국 사람의 직관에 의지할 수밖에 없다는 한계가 있다.

이와는 별도로 최근 프로세스 마이닝과 인공지능 기술을 융합하려는 시도가 여러 차례 있었다. 머신러닝 알고리즘을 통해 프로세스의 흐름을 분석하여 다음 단계의 프로세스, 또는 완료될 때까지의 수행 시간을 예측하는 연구들은 프로세스가 완료되기 전에 예측 모델을 통해 이슈가 발생할 가능성이 높은 Case들을 찾아내고 이를 사전에 관리하여 업무를 효율화하기 위한 목적으로 활용된다. 이런 예측 모델은 프로세스 마이닝 도구의 새로운 활용 방안을 제시하였으나 프로세스 마이닝의 핵심 목적인 이슈에 대한 원인분석 자체에는 도움이 되지 못하고 있다.

설명 가능한 인공지능은 블랙박스 형태인 머신러닝의 구조를 분석하여 “인공지능이 예측한 결과에 대해 사람이 이해할 수 있는 수준으로 설명하는 것을 목적” (Gunning, 2017)으로 개발되었으나, 만일 특정한 머신러닝 모델이 충분히 신뢰할 만한 예측 결과를 보여준다면 반대로 어떤 특성이 해당 결과에 영향을 주었는지를 판단하는 데도 사용할 수 있다. 즉 고양이와 강아지

를 구분하는 인공지능 모델에 설명 가능한 인공지능을 통해 분석한 결과 각 사진에서 귀와 코에 대한 부분이 결과에 가장 큰 영향을 끼쳤다면 귀와 코의 형태가 개와 고양이를 구분하는 중요한 요소라는 결론을 얻을 수 있다.



[그림 1-2] 프로세스 원인분석 자동화 발전 단계

[Fig. 1-2] Process Causal Analysis Automation Development Stage

이런 설명 가능한 인공지능의 아이디어를 프로세스 마이닝 예측 모델에 적용하면 분석하고자 하는 업무 프로세스의 이슈를 정의한 후 프로세스 예측 모델을 통해 예측했을 때 충분히 신뢰할 만한 예측 정확도를 보이는 모델에 대해 설명 가능한 인공지능을 통해 결과에 영향을 끼치는 특성을 파악한다면 결국 업무 프로세스의 이슈에 대한 원인을 자동으로 발견해낼 수 있다는 가설을 세울 수 있다. 결국 [그림 1-2]에서 볼 수 있듯이 최초에 사람이 수행하던 프로세스 혁신 방법론에 데이터 마이닝의 각종 통계 및 데이터 분석 기법을 적용하여 프로세스 마이닝이라는 새로운 개념이 탄생하였고, 프로세스 마이닝과 머신러닝 알고리즘을 융합함으로써 프로세스 예측 모델을 개발하여 사전에 프로세스 수행 결과를 예측할 수 있게 된 것처럼, 프로세스 예측 모델에 다시 설명 가능한 인공지능 알고리즘을 융합함으로써 최종적으로 사람의

개입 없이 업무 프로세스의 이슈에 대한 원인을 찾아내는 ‘설명 가능한 인공지능 기반의 프로세스 이슈 원인분석 자동화’로 단계적으로 발전하였다.

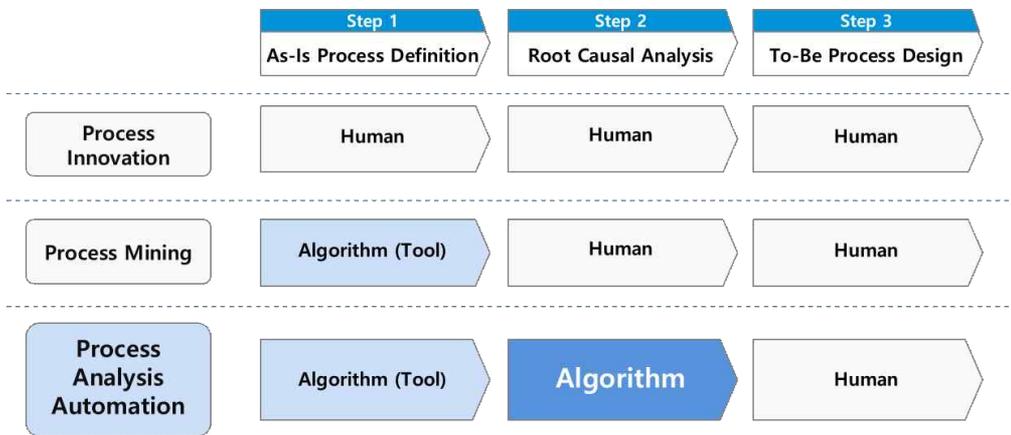
다시 말하면 프로세스 이슈 원인분석 자동화를 연구하기 위해서는 프로세스 혁신 방법론, 데이터 마이닝, 머신러닝, 프로세스 예측 모델, 설명 가능한 인공지능에서 사용되는 기법과 알고리즘을 모두 사용해야 한다.

## 제 2 절 연구의 목적

업무 프로세스 혁신 프로젝트는 크게 현행 프로세스 정의 후, 이슈에 대한 근본 원인을 분석하여 목표 프로세스 모델을 설계하는 과정으로 수행된다. 이 중 현행 프로세스 모델을 작성하는 과정은 보통 전문가가 담당자 인터뷰를 통해 정의하였으나 최근에는 업무 대부분이 시스템을 통해 진행되기 때문에 실제 업무 담당자도 컴퓨터에 입력하는 내용만 알뿐 업무 수행 절차에 대해 제대로 파악하지 못하는 일도 있고, 코로나 19의 영향으로 비대면 업무가 증가하여 실질적인 인터뷰가 힘든 문제가 발생했다. 이에 프로세스 마이닝 도구를 사용하면 [그림 1-3]와 같이 기존의 정보시스템 로그 데이터에서 자동으로 현행 프로세스 모델을 생성해주므로 더욱 빠르고 정확하게 현행 프로세스 정의가 가능하다.

그러나 발견된 현행 프로세스의 문제점에 대한 ‘원인분석’은 여전히 전문가의 영역에 머물러 있다. 본 연구에서 제안된 ‘설명 가능한 인공지능 기반의 프로세스 이슈 원인분석 자동화’는 [그림 1-2]처럼 기존의 프로세스 마이닝에서 자동화하지 못한 원인분석 단계를 알고리즘 기반으로 자동화함으로써 더욱 빠르고 정확하게 업무 프로세스 혁신이 가능하도록 지원할 것이다. 지금까지 전문가의 참여 없이는 불가능하다고 여겨지던 프로세스 이슈 원인분석을 알고리즘만으로 자동화하고, 이를 통해 지금까지 전문가의 경험과 개인 역량에 의존하던 업무 프로세스 혁신을 위한 문제 분석 과정을 자동화함으로써 분석인력의 역량과 상관없이 같은 결과를 도출해 낼 수 있게 될 것이다. 또한, 그동안 많은 인력과 시간이 소요되고도 그 결과를 확신할 수 없었던 기업

의 업무혁신 프로젝트를 훨씬 적은 인력과 시간으로 더 정확히 진행할 수 있게 될 것이다. 또한, 사람의 감에 의한 분석이 아닌 수치에 의한 정량적인 분석이므로 더욱 정확한 분석과 그 결과에 대한 근거 제시가 가능할 것이다. 나아가 프로세스 마이닝 솔루션의 기능으로 구현하여 시스템화된다면 전문가의 참여 없이도 상시적인 이슈 파악 및 이슈에 대한 원인분석까지 가능하므로, 기업은 지속적인 업무혁신이 가능하고 이를 통해 경영성과를 달성하고 지속해서 성장하는 데 도움을 줄 수 있을 것이다.

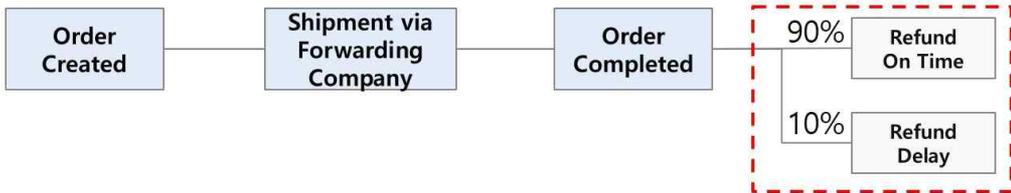


[그림 1-3] 프로세스 혁신 자동화 단계  
[Fig. 1-3] Process Innovation Automation Stages

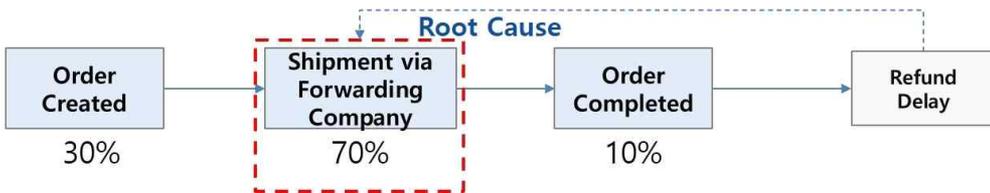
본 연구에서 제시하는 ‘설명 가능한 인공지능 기반의 프로세스 이슈 원인 분석 자동화’는 기존 인공지능 기술의 이용과는 차별화된 새로운 활용 방안을 제시하고 있다. 기존의 머신러닝과 딥러닝 등 인공지능 기술을 활용하는 것은 주로 이미지, 음성 등에 대한 인식 분야와 예측 분야에 한정되어 있었다. 프로세스 마이닝 분야에서 인공지능 기술을 사용할 때도 지금까지는 다음 프로세스를 예측하거나 총 소요 시간을 사전에 예측하는 데에 초점이 맞춰져 있었다. 그러나 본 연구는 기존의 머신러닝 기반 프로세스 예측과는 상이한 접근 방법을 취한다. [그림 1-4]와 같이 기존 프로세스 예측 모델 연구에서는 Order Created → Shipment via Forwarding Company → Order Completed의 3개 프로세스가 수행되었을 때 업무 처리가 정시에 끝날 것인

가 또는 지연될 것인가를 ‘예측’하는 데에 초점을 맞추었다면 본 연구는 예측 모델에 의해 ‘지연’이라는 결과가 예측되었을 때 결과에 가장 큰 영향을 준 요인이 “Shipment via Forwarding Company”라는 것을 찾아내는 것이 목적이다.

1) 머신러닝 기반의 예측(Prediction)



2) 설명가능한 인공지능 기반의 원인 분석(Causal Analysis)



[그림 1-4] 프로세스 예측과 원인분석의 차이 (Jung, & Lee, 2020)  
 [Fig. 1-4] Difference between Process Prediction and Process Causal Analysis Automation

또한, 기존의 설명 가능한 인공지능 알고리즘은 이미지 인식 등에서 인공지능이 해당 객체를 파악하는 과정을 사람에게 설명하기 위한 용도로 많이 쓰였다. 그러나 이를 프로세스 마이닝과 같이 결과 예측보다는 원인분석이 더 중요한 업무에 활용한다면 원인분석 자동화의 용도로 확장 가능할 것이다.

제 3 절 연구 범위 및 논문의 구성

1) 연구의 범위

프로세스 분석 자동화를 위하여 프로세스 마이닝 기반의 로그 데이터 수

집 및 정제작업이 필요하다. 또한, 해당 데이터를 기반으로 전문가의 분석을 통해 여러 가지 프로세스 이슈에 대한 원인분석이 되어 있어야 한다. 이를 위해 국내외의 다양한 프로세스 마이닝 프로젝트 사례 및 데이터를 조사하였으며 그중 공개가 가능한 일부 데이터 세트를 사용하였다. 해당 데이터 세트에 관한 내용을 이해하고 프로세스를 시각화하기 위하여 프로세스 마이닝 도구 및 관련 기법을 활용하였으며, 데이터 세트를 기반으로 최종 프로세스 수행 결과를 예측하기 위해 다양한 머신러닝 및 딥러닝 관련 기법을 활용하였다. 특히 예측 결과에 대한 원인분석의 핵심 기술로 설명 가능한 인공지능 알고리즘을 사용하였으며, 다양한 기법들을 실험하여 알고리즘별 특징을 파악하고자 하였다. 전체적으로 프로세스 마이닝과 머신러닝을 이용한 프로세스 예측 모델, 그리고 설명 가능한 인공지능 알고리즘의 기술요소들을 모두 활용하여 프로세스 이슈 원인분석 자동화를 위한 프로그램을 개발하여 실험을 진행하였다.

기존의 프로세스 마이닝 도구들은 이미 상용화된 솔루션으로 다양한 사용자 인터페이스를 제공하고 있으며 범용적으로 사용할 수 있다. 그러나 본 연구는 이슈 원인분석 자동화를 위한 알고리즘을 검증하는 데 목적이 있으므로 별도의 사용자 화면을 개발하지는 않았다. 또한, 사용 데이터와 목적에 맞춰 각각의 프로그램을 개별적으로 개발하였기 때문에 범용적인 솔루션이 아니다.

본 연구는 프로세스 마이닝 프로젝트의 데이터를 개인정보 삭제 후 온라인에 공개해준 관련 기업들과 GitHub에 소스 코드를 공개해준 연구자들의 도움을 받아서 수행할 수 있었으며 본 논문에서 사용된 모든 소스 코드와 가공된 데이터 역시 GitHub<sup>1)</sup>을 통해 공개하였다.

## 2) 논문의 구성

1장에 이어서 2장에서는 연구에 사용되는 현대 경영혁신 이론과 업무 프로세스 혁신 (Process Innovation)의 이론적 배경과 실제에 대해 알아보고, 프로세스 마이닝을 통해 현행 업무 프로세스 모델을 자동 생성하고 분석하는

---

1) [https://github.com/bankplus/process\\_issue\\_analyzer](https://github.com/bankplus/process_issue_analyzer)

방법의 장점과 한계를 확인하였다. 또한, 현대의 머신러닝과 딥러닝 알고리즘의 종류와 머신러닝 모델에 기반한 프로세스 예측 모델의 발전 현황을 정리하였다. 마지막으로 예측 모델 자체를 분석하기 위한 설명 가능한 인공지능 알고리즘의 개념과 다양한 알고리즘별 특징을 분석하였다.

3장에서는 기존 전문가 기반의 프로세스 이슈 원인분석 방법론과 새롭게 제안된 ‘설명 가능한 인공지능 기반의 프로세스 분석 자동화’의 차이점과 같은 데이터를 기반으로 실험하였을 때 인간이 분석한 결과와 동일한 결론을 내린다면 프로세스 분석을 자동화할 수 있다는 연구 가설을 다시 한번 설명하고, 실험에 사용된 연구 데이터 수집 및 변환, 다양한 머신러닝 모델 및 설명 가능한 인공지능 알고리즘을 테스트하기 위한 연구 도구 및 개발환경, 라이브러리에 관해 설명하였다.

실험 및 분석 결과를 정리한 4장에서는 3가지 유형의 데이터에 대한 실험을 진행하였다. 정찬일, & 이후진 (2019b)에서 발표한 네덜란드의 R은행 여신 프로세스를 기반으로 대출 신청 후 고객에 의해 취소 (Cancel) 되는 Case를 사전에 예측하는 딥러닝 모델을 만들어 정확도를 측정 후 기본적인 설명 가능한 인공지능 모델을 통해 취소의 원인을 분석해보는 과정을 재정리하였다. 이 실험을 통해 사람의 개입 없이도 이슈에 대한 원인분석이 자동화될 수 있는지 검증하였다. 다음으로 정찬일, & 이후진 (2020)에서 발표한 바와 같이 글로벌 전자제품 제조기업인 P사의 환불 프로세스를 기반으로 딥러닝을 포함한 다양한 머신러닝 모델을 통해 처리시간이 지연되는 Case를 예측해보고, 학습된 예측 모델에 다시 LIME, SHAP, LRP 등 다양한 ‘설명 가능한 인공지능 알고리즘’들을 적용하여 해석 모델을 작성한 후 예측 모델과 해석 모델 각각의 특징과 장단점을 비교하여 업무 프로세스 이슈에 대한 원인분석을 위한 최적의 방법을 모색하였다. 비교 분석 결과 딥러닝 모델의 예측 정확도가 다소 높지만, 원인분석 결과의 정확도는 전통적인 머신러닝 모델들이 상대적으로 높았으며, 설명 가능한 인공지능 알고리즘 간의 정확도 차이는 크지 않다는 것을 확인하였다. 그러나 알고리즘 간의 차이보다는 학습데이터의 전처리와 구조를 어떻게 구성하는지가 결과의 정확도에 가장 큰 영향을 준다는 것을 알 수 있었다. 마지막으로 Jung, & Lee (2020)에서 발표한 한국의 모바

일 음원 스트리밍 서비스 기업인 M사의 프로세스 마이닝 사례를 재정리하여 프로세스 데이터를 새로운 방식으로 전처리하고 다양한 속성들을 함께 활용하여 음원 이용권 결제 실패 Case를 예측하였으며 알고리즘에 의해 자동으로 분석된 결제 실패의 원인이 프로세스 마이닝 프로젝트를 통해 전문가들이 분석한 결제 실패의 근본 원인과 동일함을 증명하여 프로세스 이슈 원인분석 알고리즘은 실제 업무혁신 프로젝트에서도 충분히 활용 가능한 수준의 정확도를 확보하였다.

마지막으로 5장에서는 전체 논문의 내용을 요약하여 설명하고 연구의 기여와 한계, 향후 발전 방향 등에 대해 정리하였다.

## 제 2 장 선행연구 및 한계

### 제 1 절 경영 혁신과 프로세스 마이닝

#### 1) 현대 경영과 프로세스 혁신

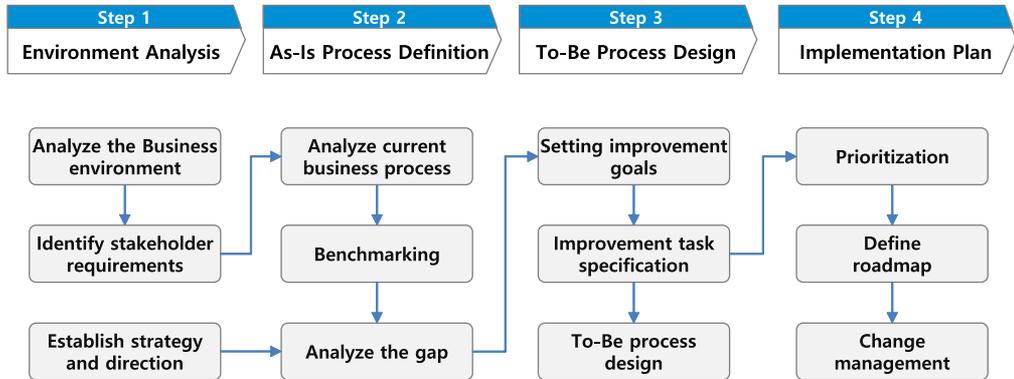
현대 경영학의 대가(大家)로 불리는 Drucker (1998)는 경영혁신은 “기존의 지식, 제품, 고객의 요구, 시장 등에서 부족한 점을 발견하여 새롭고 더 생산적인 것으로 변화시키는 일”이라고 정의하고 “비즈니스 모델 혁신과 기술혁신을 통한 제품혁신, 그리고 이 두 가지 혁신의 근간이 되는 프로세스 혁신 (Process Innovation)”의 중요성을 강조했다.

또한, 대부분의 경영 이론에서도 프로세스 혁신을 위한 절차 및 방법론을 핵심으로 제시하였다. 예를 들어 Six Sigma에서는 DMAIC (Define-Measure-Analyze-Improve-Control) 방법론을 정의하였고 (Pyzdek, & Keller, 2014), CPM (Corporate Performance Management)에서는 전략/목표설정 (Strategy Formulation), 시나리오 분석 (Scenario Analysis), 계획과 예산 수립 (Plan & Budget), 의사소통 (Communicate), 성과측정 (Performance Monitoring), 예측 (Forecasting), 보고서 작성 (Report)의 절차를 정의했다 (Bourne, Franco, & Wilkes, 2003). 또한, CPI (Continuous Process Improvement)는 기업의 지속적인 프로세스 개선을 위해 SAMIE (Select-Analysis-Measure-Improve-Evaluate) 방법론을 사용하였으며, (Robson, 2010), TQM (Total Quality Management)에서도 8가지 원칙 (Customer Focus, Total Employee Commitment, Process Approach, Integrated System, Strategic and Systematic Approach, Continual Improvement, Fact-based Decision-making, Communications)에 Process Approach를 포함하였다 (Dahlgard, Khanji, & Kristensen, 2008).

특히 “비용, 품질, 서비스, 속도와 같은 핵심적 성과에서 극적인 향상을 이루기 위해 업무 프로세스를 근본적으로 다시 생각하고 혁신적으로 재설

계”(Hammer, 1990) 하는 BPR (Business Process Reengineering) 방법론과 이를 기반으로 정보기술의 통합 및 ERP (Enterprise Resource Planning) 구축 등을 통하여 업무 프로세스, 정보시스템 및 조직 등을 총체적으로 혁신함으로써 경영성과를 높이려는 PI (Process Innovation) 방법론 (Davenport, 1993)은 현재도 많은 기업에서 활발하게 추진되고 있는 주요한 경영기법으로 프로세스 혁신을 가장 중요하게 다루고 있다. 특히 빠르게 변화하는 시장과 고객, 업종의 구분이 없이 치열해지는 경쟁 등 최근의 경영환경에서는 기업의 혁신과 디지털화가 화두로 대두되고 있다. 세계 1위의 휴대전화 기업이었던 노키아조차도 변화의 물결에 올라타지 못하고 자신의 OS만을 고집하다가 채 10년도 되지 않은 시간에 폐업 위기에 놓였다가 헐값에 매각되었다. 이러한 시대의 조류에 대응하기 위해 많은 기업이 혁신을 화두로 삼고 있으며 이러한 업무혁신과 디지털 전환 (Digital Transformation)을 위한 주요한 도구로서 PI 프로젝트를 반복적으로 수행하고 있다. PI를 수행할 만한 경제적인 여력이 되는 기업 대부분은 보통 3~5년에 한 번 이상 PI를 수행하고 있다.

그러나 Cozijnsen, Vrakking, & van IJzerloo (2000)이 네덜란드의 혁신 프로젝트 (Innovation Project)들을 조사한 결과를 살펴보면 전체 프로젝트의 20~30%가량만이 성공할 뿐이었다. 이렇게 혁신 프로젝트가 성공하지 못하는 주요 원인은 명확한 목표설정의 실패, 컨센서스 미확보, 적절한 인력 (자원) 배치 실패, 변화관리의 문제, 경영진의 리더십 문제 등으로 파악되었다. Cozijnsen, Vrakking, & van IJzerloo (2000)가 나열한 실패의 원인들의 공통점은 모두 사람의 문제라는 점이다. 적절한 인력 배치란 결국 경험이 많은 내부 인력과 전문성을 가진 외부 컨설턴트가 참여해야 한다는 것이고, 변화관리와 경영진의 리더십은 조직 문화를 개선해야 한다는 의미로 해석할 수 있다. 이는 실패 원인에 관한 연구 자체도 프로젝트를 수행한 참여 인력과의 인터뷰와 설문을 통해 이루어졌기 때문이라고 할 수 있다. 이를 다른 관점에서 보면 결국 근본적으로는 참여 인력의 경험과 개인적인 역량에 의존하는 데에 있다고 볼 수 있다. 다만 대부분의 혁신 프로젝트는 모두 경험 있는 전문가들의 주도로 진행되고, 만일 참여 인력의 역량이 모자란다면 프로젝트의 성공 확률은 그만큼 줄어들 수밖에 없다는 것이 현실이다.



[그림 2-1] PI 프로젝트 수행 방법론 예시 (정찬일, & 이후진, 2019a)  
 [Fig 2-1] PI Project Methodology Example

현재 금융회사 및 제조 기업에서 진행되고 있는 PI 프로젝트의 사례를 보면 일반적으로 [그림 2-1]과 같이 환경분석 (Environment Analysis), 현행 프로세스 정의 (As-Is Process Definition), 목표 프로세스 설계 (To-Be Process Design), 이행계획 수립 (Implementation Plan)의 절차로 진행된다. 이때 현행 업무 프로세스의 파악은 대부분 담당자와 임직원에게 대한 인터뷰를 통해 진행된다 보니 정성적인 수준의 기대효과만 제시하게 되어 현업 및 경영진의 공감을 얻기 어려운 문제가 있다. 또한, 목표 업무 프로세스의 설계는 컨설턴트 또는 몇몇 전문가 개인의 역량에 지나치게 의존하게 되는 문제가 발생한다.

이러한 문제에 대한 인식에 기반하여 실제 컨설팅 업계에서는 PI 프로젝트를 개선하기 위해 다양한 방법을 연구해왔다. 최근에는 지속적으로 관리 가능한 업무혁신 (Sustainable PI)를 위해 프로세스 거버넌스 체계를 구축하기 위한 노력이 진행되고 있다. 이는 PI를 수행하기 위한 인력과 비용이 많이 들어 자주 수행하기 어려울 뿐 아니라 참여 인력의 수준에 따라 성공 여부가 결정되는 위험을 낮추고 내부 인력 중심으로 지속적인 업무혁신을 수행하기 위한 방법론이다.

무엇보다 “측정하지 못하면 관리할 수 없고, 관리하지 못하면 개선할 수 없다”라는 Drucker (1998)의 유명한 말처럼 정량적으로 측정되지 못하고 사

람의 직관에 따른 판단을 하는 것 자체가 이미 상당한 위험을 내포하고 있다고 볼 수 있다. 이에 대한 대안으로 데이터웨어하우스와 데이터 마이닝 등의 다양한 분석 방법들이 대두되었으나 업무혁신 방법론과 접목하기에는 접근 방식에서 차이가 나기 때문에 활용되지 못하고 있던 상황에서 데이터 마이닝 기술과 프로세스 혁신을 융합한 프로세스 마이닝 기술이 두각을 나타내기 시작했다.

## 2) 프로세스 마이닝을 통한 프로세스 모델 생성

프로세스 마이닝은 기업의 업무혁신을 위해 데이터 마이닝의 다양한 기법을 융합한 것으로 데이터 마이닝과 유사하지만, 데이터가 아닌 프로세스 자체에 집중한다는 점에 특징이 있다. 데이터 마이닝이 매출액, 손익 등과 같이 프로세스 수행의 최종 결과 중심으로 분석한다면, 프로세스 마이닝은 전체 프로세스 (End to End Process)의 수행 과정 전체를 분석하고 시각화하여 어떻게 공정을 개선할 것인지에 집중한다. 이러한 프로세스 마이닝은 제조업의 공정 자동화를 위한 프로세스 분석부터 금융, 서비스, 물류 등 다양한 업무에 적용되며 최근에는 인터넷 서비스의 로그를 분석하는데에도 활용되고 있다.

프로세스 마이닝은 정보시스템에서 업무를 수행할 때 기록된 이벤트 로그를 기반으로 [표 2-1]처럼 ①프로세스를 발견 (Process Discovery)하여 모델을 자동 생성하고 실제 수행된 프로세스가 표준 프로세스 모델에 기반하였는지에 대한 ②적합성 평가 (Conformance Checking)를 하여 ③프로세스 모델을 확장·개선 (Enhancement)하는 기법이다(Van Der Aalst, 2011). Process Discovery란 기존의 업무 로그를 조합하여 프로세스 모델을 생성해내는 것이다. 이렇게 생성된 프로세스 모델과 기존에 업무 규정, 매뉴얼 등에서 정의해 놓은 이론적인 프로세스가 서로 일치하는지 확인하는 작업을 Conformance Checking이라고 한다. 이러한 비교 작업을 통해서 현행 프로세스의 문제를 찾고 개선 방향을 도출한다면 이를 통해 업무 규정과 매뉴얼을 개선하고 프로세스 모델을 개선할 수 있다. 이러한 작업을 Enhancement라고 말한다.

[표 2-1] 프로세스 마이닝 구성 요소  
 [Table 2-1] Components of Process Mining

Classification	Contents	Tools
Process Discovery	Based on event logs, process model is constructed or discovered	Alpha Algorithm
Conformance Checking	The process model is compared with the system event log; Inconsistencies between log and model are analyzed	Decision Tree
Enhancement	The model is extended with additional performance information; improve the performance of the existing model	-

프로세스 마이닝을 활용하기 위해서는 정보시스템의 Event Log 데이터는 Case ID, Activity, Timestamp가 필수 요소이다. 예를 들어 [표 2-2]와 같은 ‘제품 주문’ 프로세스가 있다고 하면, 제품코드 LT9123은 2021.05.02 10:20:31에 주문 계약을 등록한 후 Sale Order, Delivery Order, Good Issue의 순서로 프로세스가 진행된다. 이때 각각의 제품별 주문진행 1건을 Case라고 하고 이 Case는 서로 다른 ID를 가진다. 물론 여러 개의 제품을 한 번에 주문한다면 Case ID 하나에 여러 제품이 등록될 수도 있다. 또한, 각각의 ‘Sales Order’ 등록, ‘Delivery Order’ 등록과 같은 업무 활동을 Activity라고 한다. 이때 각각의 Activity는 Timestamp가 명시되어야 하는데, 이는 Activity 간의 순서를 자동으로 만들기 위해서이다. 또한, 일반적으로 해당 업무를 수행한 사람 또는 시스템을 Resource라고 하며 그 외에도 제품번호, 고객 코드 등의 다양한 속성을 추가할 수 있는데 이것을 Attribute라고 한다.

예전에는 이벤트 로그를 수집하더라도 오프라인에서 수작업으로 진행되는 업무가 많았기 때문에 기업의 전체적인 프로세스를 확인하기에는 한계가 있었다. 그러나, 최근 기업 업무의 대부분은 정보시스템을 이용하여 수행되고 애초에 인터넷이나 모바일을 이용하는 서비스 기업이 늘어나고 있다. 또한, 시스템의 로그와 같은 비정형/반정형 빅데이터의 수집 및 분석 기술도 발달

하고 있어 프로세스 마이닝의 필수 요소인 로그 데이터의 수집 및 분석이 점점 더 용이해지고 있어 프로세스 마이닝의 적용 범위와 활용도는 더욱 넓어지고 있다.

[표 2-2] 이벤트 로그 필수 요소  
[Table 2-2] Essential Element of Event Log

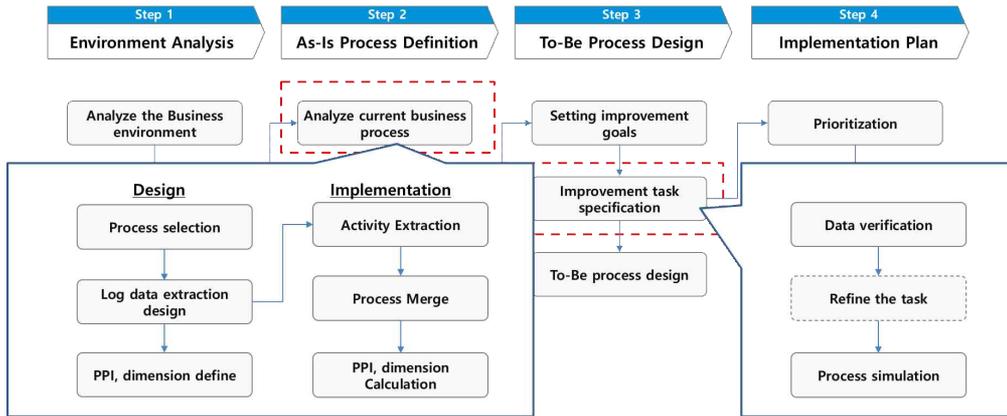
Case ID	Activity	Time stamp	Resource	Product ID
A0501-1	Contract	2021.05.02 10:20:31	11011	LT9123
A0501-1	Sales Order	2021.05.02 13:43:15	11011	LT9123
A0501-2	Sales Order	2021.05.02 15:10:15	11021	LT9125
A0501-1	Delivery Order	2021.05.03 11:10:11	11011	LT9123
A0501-1	Good Issue	2021.05.03 11:20:34	11011	LT9123
A0501-2	Create Invoice	2021.05.04 15:30:11	11021	LT9125
...	...	...	...	...

이에 IEEE에서는 Process Mining Task Force<sup>2)</sup>를 구성하여 표준화 및 활성화화를 위한 노력을 진행하고 있으며, 많은 소프트웨어 제조사들이 관련된 솔루션을 출시하고 있다 (Van Der Aalst et al., 2011). Process Mining Task Force에는 IBM, HP, Infosys, Fujitsu 등의 기업과 Gartner, Deloitte 등의 글로벌 컨설팅 기업, 약 20여 개의 대학이 참여하고 있다. 또한, 네덜란드 시청을 포함한 공기업 등의 업무 프로세스 분석, 암스테르담 종합 병원의 환자 치료 프로세스 분석 등의 업무 프로세스 분석 등에 활용되고 있다. 또한, 반도체 생산 장비 업체인 ASML은 생산 장비의 임베디드 시스템 프로세스 분석에 활용하였고, 필립스는 메디컬 장비용 소프트웨어를 테스트하기 위해 프로세스 마이닝 기법을 활용하고 있다.

국내에서도 삼성전자, 삼성전기, 대우조선해양 등 제조 기업의 제조 프로

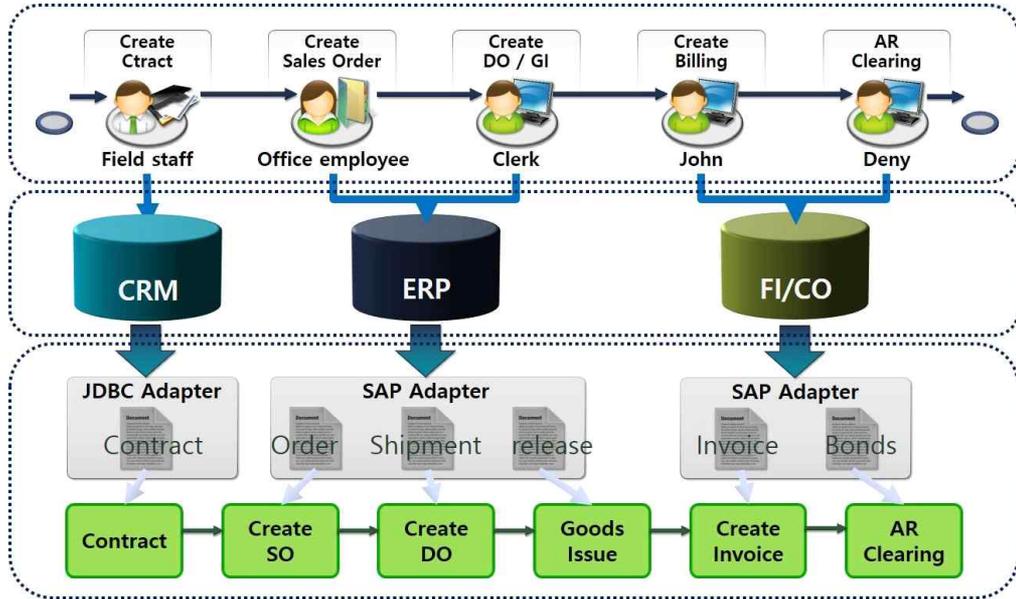
2) "The Data Mining Technical Committee (DMTC) of the Computational Intelligence Society (CIS) of the Institute of Electrical and Electronic Engineers, Inc. (IEEE)."

세스 분석, 분당서울대학교병원, 보라매 병원 등의 진료 프로세스 분석, 부산항의 항만 물류 프로세스 분석, 전시회 관람객 동선 분석 등 다양한 분야에서 프로세스 마이닝이 적용되고 있다 (송민석, 2013).



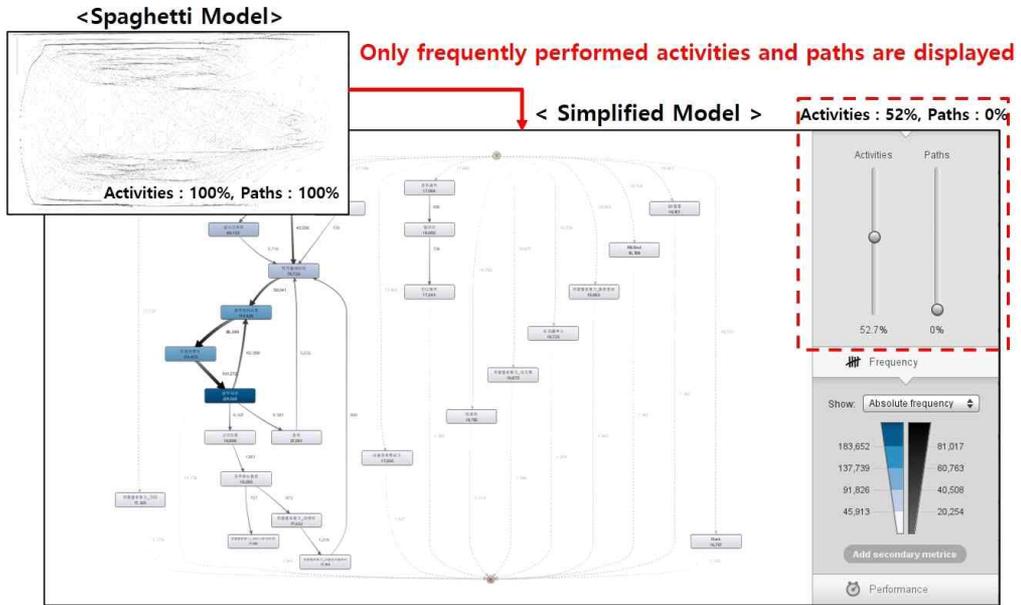
[그림 2-2] PI에서 프로세스 마이닝 적용 (정찬일, & 이후진, 2019a)  
 [Fig 2-2] Process Mining for PI Project

상기한 바와 같이 업무혁신 프로젝트에서는 최근 프로세스 마이닝을 적극적으로 도입하고 있는데 [그림 2-2]처럼 현행 업무 프로세스를 분석하는 데 주로 사용된다. 전체 프로세스를 모두 프로세스 마이닝으로 분석하기에는 시스템 로그 데이터를 수집하고 프로세스 마이닝 도구를 통해 시각화하는 데에는 한계가 있으므로 보통은 중요 프로세스를 선정하여 프로세스 모델을 자동으로 생성한 후 정량적으로 측정하고 분석한다. 현행 프로세스 분석 단계에서 프로세스 마이닝을 활용하는 절차는 크게 설계와 구현으로 구분하는데, 설계 단계에서는 대상 프로세스를 선정하고 로그 데이터 수집 방안과 각종 지표를 설계한다. 구현 단계에서는 실제 프로세스 마이닝 도구를 사용하여 Activity를 추출하고 추출된 Activity를 Merge하여 프로세스 모델을 생성하고 설계된 각종 지표를 측정하는 작업을 진행한다.



[그림 2-3] 프로세스 모델 자동 생성 (정찬일, & 이후진, 2019a)  
 [Fig. 2-3] Process Model Discovery

프로세스 마이닝을 활용하면 [그림 2-3]와 같이 다양한 정보시스템을 사용하는 업무 프로세스도 별도의 어댑터 연결만으로 관련 로그를 수집하여 통합된 프로세스 모델 형태로 도출할 수 있다. 그러므로 기존의 PI 프로젝트에서 인터뷰를 통해 현행 프로세스를 파악하는 대신 정량적인 데이터 기반으로 프로세스 모델을 생성 (Process Discovery)하는 것이 가능하다. 물론 프로세스 마이닝만으로는 수작업 프로세스는 파악되지 않는다는 한계가 있지만, 최근에는 업무 대부분이 정보시스템을 이용하고 기존에는 수작업으로 진행했던 업무들도 모바일 기기를 이용하여 즉시 결과를 등록하고 있으므로 프로세스 마이닝의 유용성은 더욱 높아지고 있다. 특히 예전의 업무 프로세스 혁신 프로젝트는 아직 정보화되지 않은 분야를 찾아서 정보시스템을 도입하는 것만으로도 충분한 혁신과 생산성 향상을 기대할 수 있었다면, 최근의 기업들은 오히려 정보화되지 않은 분야를 찾기 어렵고 기존의 정보시스템에서 해결되지 못하는 문제들을 찾는 것이 주요 활동이 되었다. 이런 경우 기존의 정보시스템 로그를 기반으로 현행 프로세스를 작성하여 문제를 찾아내는 프로세스 마이닝은 특히 유용하다.



[그림 2-4] 프로세스 마이닝 모델의 복잡성 (강영식, & 이보경, 2016)  
 [Fig. 2-4] Complex of Process Mining Model

그러나 프로세스 마이닝 도구를 통해 자동으로 생성된 프로세스 모델은 너무나도 복잡하여 사람의 육안으로는 확인할 수 없으므로 프로세스 마이닝 도구인 ProM<sup>3)</sup>, DISCO<sup>4)</sup>, PPM<sup>5)</sup> 등에서 제공하는 다양한 필터 기능을 이용하여 프로세스를 추상화하는 작업을 수행해야 한다. [그림 2-4]과 같이 이벤트 로그에서 자동으로 생성된 프로세스 모델은 너무 복잡해서 사람의 눈으로는 확인할 수 없는 스파게티 모델이 되므로 그중 가장 자주 출현하는 Activity만으로 필터링하여 프로세스 맵을 단순화하여야 한다 (강영식, & 이보경, 2016).

추상화된 모델의 경우 사람의 직관을 통한 분석에는 유리하나 주로 문제가 되는 예외 프로세스를 놓치게 되고 데이터가 왜곡되어 정확한 분석에 한

3) ProM은 플러그인 형태로 다양한 프로세스 마이닝 기술을 지원하는 확장 가능한 오픈소스 프레임워크 (<http://www.pmtools.org>)

4) DISCO는 Fluxicon 사의 프로세스 마이닝 도구로 프로세스 맵을 도출하는 데 강점

5) PPM은 SoftwareAG사의 프로세스 마이닝 도구로 ERP 및 기업의 프로세스 분석 기능에 강점

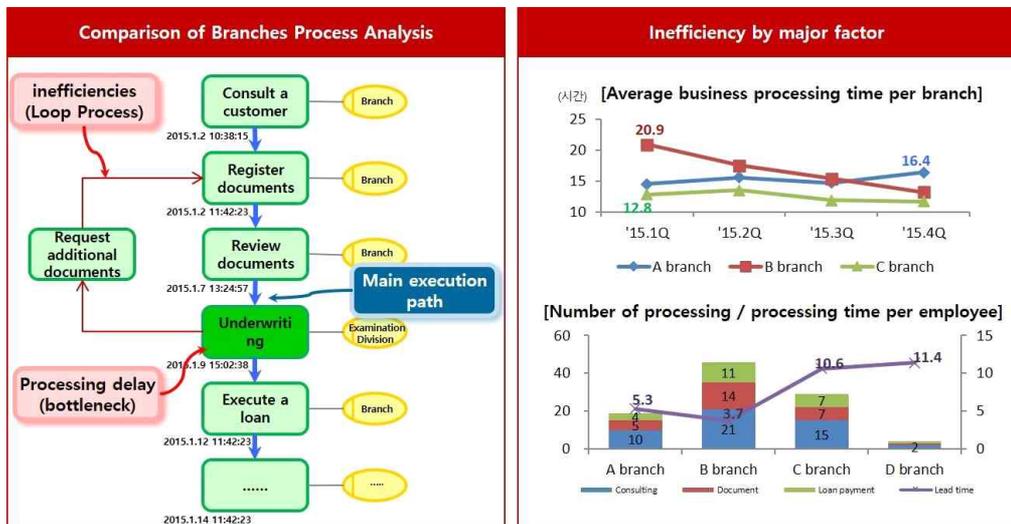
계가 발생할 수 있다. 또한, 프로세스 마이닝 도구들은 다양한 분석 기능 (필터 등)을 제공하고 있으나 명확한 패턴이 정해져 있는 것이 아니라 다양한 분석 기법들을 전문가의 경험과 업무 도메인에 대한 지식 기반으로 [표 2-3]에 정의된 Process Flow, Organization, Performance (Time), KPI (Key Performance Index), Data 등 여러 가지 기준에 적용하면서 반복해서 분석해 나가는 방식을 사용하고 있다.

[표 2-3] 프로세스 분석 기법 (정찬일, & 이후진, 2019a)  
[Table 2-3] Process Analysis Method

Target	Method	Expected result
Process Flow	<ul style="list-style-type: none"> <li>• Comparison of the process model with the actual process</li> </ul>	<ul style="list-style-type: none"> <li>• Discover inefficient processes (delay, rework, bottleneck)</li> </ul>
Organization	<ul style="list-style-type: none"> <li>• Organization relation map</li> </ul>	<ul style="list-style-type: none"> <li>• R &amp; R redundancy</li> <li>• Workload by organization</li> </ul>
Performance (time)	<ul style="list-style-type: none"> <li>• Performance of Process Flow</li> </ul>	<ul style="list-style-type: none"> <li>• Find Long-Time Activity</li> </ul>
KPI, PPI	<ul style="list-style-type: none"> <li>• KPI measurement</li> <li>• PPI (Process Performance Index)</li> </ul>	<ul style="list-style-type: none"> <li>• Choosing a process for improvement</li> <li>• Root cause analysis</li> </ul>
Data	<ul style="list-style-type: none"> <li>• Data accuracy</li> <li>• Risk analysis, Outlier</li> </ul>	<ul style="list-style-type: none"> <li>• Non-compliant rules</li> <li>• Outlier condition</li> </ul>

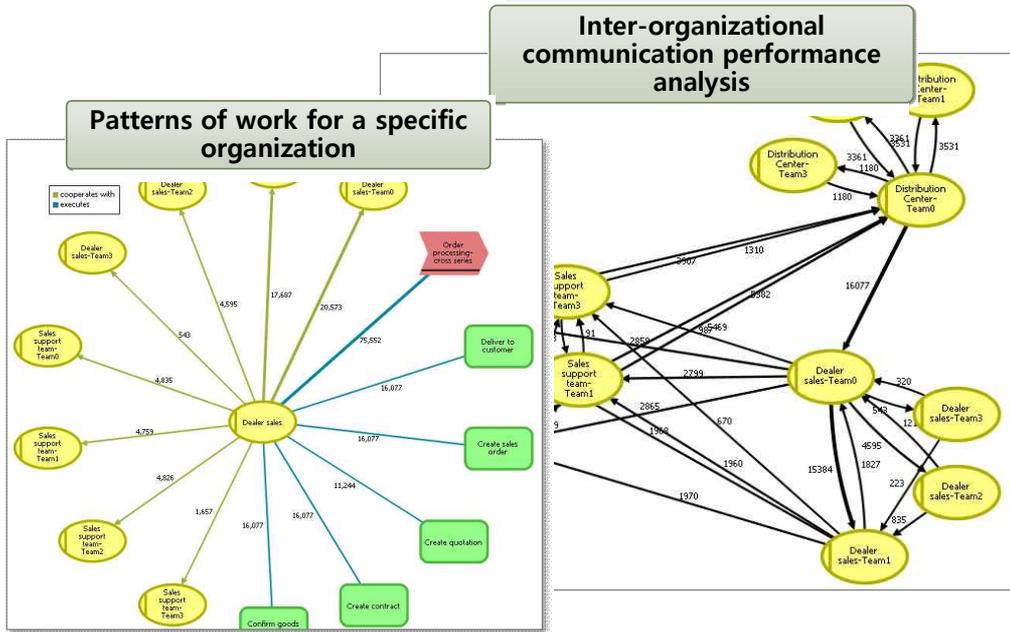
먼저 Process Flow 분석은 생성된 프로세스 모델을 확인하여 가장 시간이 오래 걸리는 프로세스 흐름을 찾는다. 일반적인 프로세스 마이닝 도구들은 프로세스 흐름의 화살표 굵기나 색깔 등으로 수행 시간을 다르게 표시한다. 이때 주의할 점은 하나의 Activity가 수행된 후 다음 Activity가 수행될 때까지 시간이 오래 걸린다고 해서 모두 지연인 것은 아니라는 점이다. 예를 들어 각 Activity 사이에 오프라인으로 처리하는 수작업 업무들이 존재한다거나, 업무

절차상 월말, 연말에만 처리해야 하는 프로세스 등이 있다면 유의해야 한다. 또한, 주의해서 확인해야 할 것은 프로세스의 반복 현상 (Loop)이다. 주로 결제, 점검 등에서 발생하는데 [그림 2-5]와 같이 적격심사 (Underwriting) 시에 추가적인 서류 징구 또는 보완이 필요하여 여러 번의 재작업이 일어난다면 최초 서류 제출 시 자동 체크하는 기능이 없거나 업무 담당자가 필요한 서류에 대한 업무 이해가 낮아서 발생하는 경우가 많으므로 서류 등록 시스템에 자동 체크 기능을 개발하거나 담당 직원들에 대한 업무 교육을 진행할 필요가 있다.



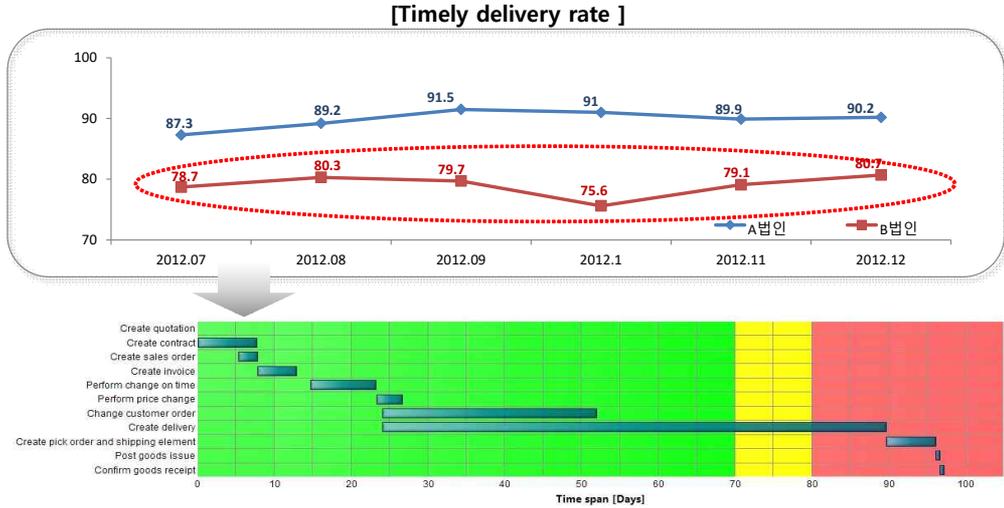
[그림 2-5] 프로세스 마이닝 분석 예시 (정찬일, & 이후진, 2019a)  
 [Fig. 2-5] Method of Process Mining Analysis

두 번째로 Organization 분석은 주로 업무 부서 간의 흐름을 중심으로 확인한다. 이는 일반적인 프로세스 모델이 아닌 [그림 2-6]과 같이 조직 간의 업무 흐름을 중점적으로 표시하는 별도의 모델을 생성해서 조직 간의 커뮤니케이션 시간, 건수 등을 활용해 Performance를 측정하고 조직간 업무 활동에서의 특정 패턴이 존재하는지 등을 검토한다.



[그림 2-6] 조직 분석 예시  
 [Fig. 2-6] Organization Analysis

다음으로 Performance 분석은 일반적으로 앞서 설명한 Process Flow가 프로세스의 반복이나 병목현상 등을 주로 보는 데 반해 [그림 2-7]과 같이 특정한 유형별로 전체 업무 수행 시간 (Lead Time)을 중심으로 상세히 비교한다. 특히 Lead Time이 오래 걸린 프로세스에 대해서 각각의 Activity 별 수행 시간 등을 그래프로 만들어 특정 프로세스에 문제가 발생하거나 대기 시간이 길어지는 낭비가 없었는지를 확인하는데 프로세스 마이닝 도구에서 자체적으로 해당 기능을 제공하기도 하지만 다른 데이터 마이닝 도구를 함께 사용기도 한다.

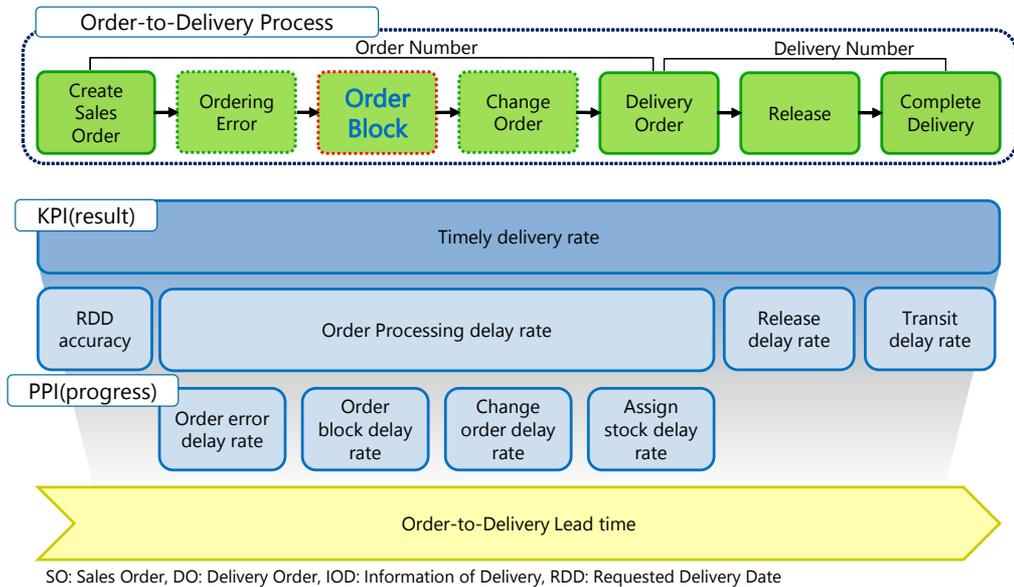


[그림 2-7] 프로세스 성능 분석 예시  
[Fig. 2-7] Process Performance Analysis

마지막으로 KPI는 기업의 목표를 정량적으로 정의한 지표로 예를 들면 ‘매출액 300조 달성’, ‘영업이익 6조원’과 같은 것들이다. 이러한 KPI의 달성을 측정한 것은 정보계 시스템을 이용하면 어렵지 않다. 그러나 대부분 기업에서 문제가 되는 것은 KPI를 달성하기 위해서 구체적으로 어떤 행동을 해야 하는지를 알 수 없다는 것이다. 매출액과 영업이익을 늘리자는 것은 실제 업무를 수행하는 말단 직원들의 입장에서는 너무 추상적인 구호에 불과하다. 프로세스 마이닝 분야에서는 KPI를 프로세스 단위로 잘게 쪼개어 관리하는 PPI (Process Performance Index)라는 용어를 더 많이 사용한다. 자동차의 무게를 1kg 줄이려고 한다면, 엔진을 뜯어내는 것이 아니라 영역별로 파트를 나누어 1g씩 줄이는 것이 더 효과적이다.

[그림 2-8]은 A 제조 기업에서 수행된 ‘적기 납기율’을 개선한 사례로 다소 추상적으로 보이는 ‘적기 납기율’ (Timely Delivery Rate)이라는 KPI를 각각의 단위 Activity 별로 나누어 측정하고 주요 지연 원인이 Order Block (주문한 기업이 신용도에 문제가 있거나 기존의 소액 연체가 남아 있는 경우, 별도로 책임자의 결재를 받도록 하는 절차)에서 발생함을 분석한 사례이다. 이 사례에서는 Order Block의 기준 (Rule)을 변경하여 우량한 기업일 경우 일정

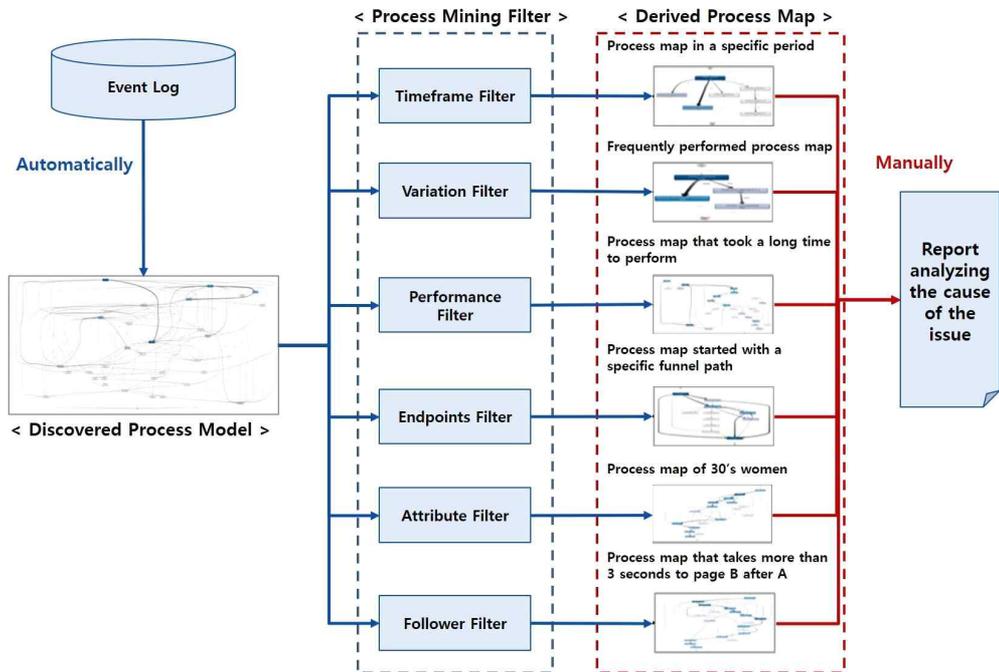
금액 이하의 소액 연체에 대해서는 추가적인 결제 절차를 제외함으로써 전체적인 KPI를 달성하였다. 이처럼 전체 KPI를 다시 프로세스와 Activity 단위로 잘게 쪼갠 후 PPI를 통해 개별 Activity의 효율성을 측정하고 이를 통해 구체적으로 행동할 수 있는 해결방안을 찾아내는 것이야말로 일반적으로 KPI 중심으로 기업의 실적을 관리하는 데이터웨어하우스나 데이터 마이닝과 같은 분석 기법과 프로세스 마이닝의 근본적인 차이점이라고 할 수 있다.



[그림 2-8]. KPI와 PPI의 관계 (정찬일, & 이후진, 2019a)  
[Fig. 2-8] KPI & PPI

대부분의 프로세스 마이닝 도구 (Tool)는 이러한 프로세스 분석을 위한 다양한 기능을 제공하고 있다. 예를 들어 DISCO는 [그림 2-9]와 같이 다양한 Filter를 제공하고 있는데 Timeframe Filter는 특정한 시간을 지정하여 해당 기간의 프로세스만 확인할 수 있는 기능, Variation Filter는 빈번하게 수행된 프로세스를 확인하는 기능, Performance Filter는 수행 시간이 오래 걸린 프로세스를 찾는 기능, Endpoints Filter는 특정 유입경로로 시작된 프로세스를 찾는 기능, Attribute Filter는 특정한 속성 (30대 이상 여성 고객)을 기준으로 프로세스를 확인하는 기능, Follower Filter는 A가 실행된 후 특정 시

간이 지나간 후에 B가 실행된 경우를 찾아내는 기능이다. 이처럼 다양한 필터 기능을 제공하는 근본적인 이유는 바로 복잡한 스파게티 프로세스를 사람이 이해할 수 있는 수준으로 추상화하기 위한 것이다. 이외에도 PPM과 같은 도구들은 일부 데이터 마이닝 기능을 추가하여 각 Activity와 Attribute에 대한 표준편차와 사전에 측정할 지표를 정의하고 이에 대한 계산 기능을 제공한다. 또한, 프로세스 마이닝 도구에서 제공하지 않는 정밀한 분석 기능이 필요할 때는 일반적으로 데이터 마이닝 도구를 함께 사용하여 분석하는 경우가 많다.



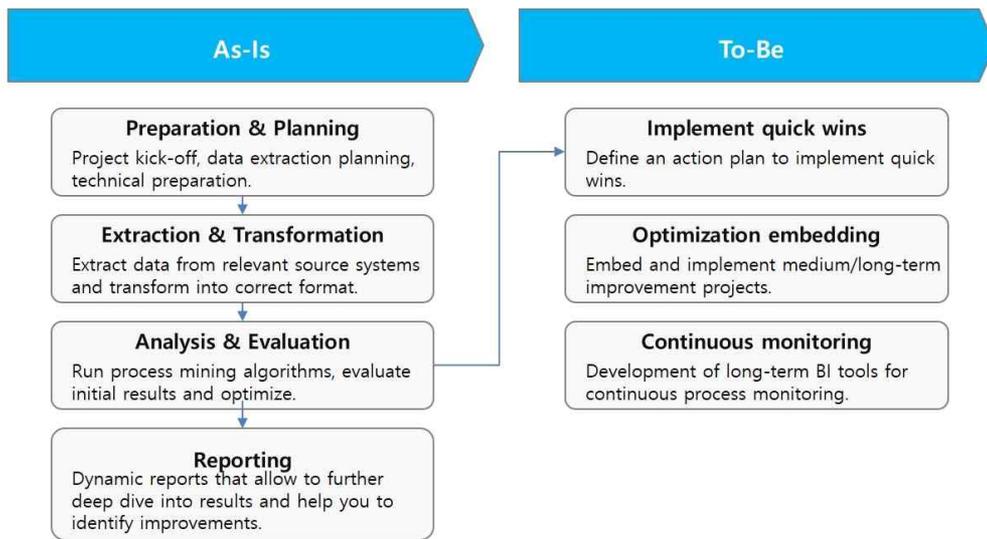
[그림 2-9] 프로세스 마이닝 분석 방법 (강영식, & 이보경, 2016; Loen Entertainment, & PMIG, 2017)

[Fig. 2-9] Method of Process Mining Analysis

이러한 프로세스 마이닝의 분석 방식은 기존의 인터뷰에 의한 업무 분석과 비교하면 정량적인 근거에 기반한 분석이 가능하다는 장점이 있다. 그러나 이것 또한 업무 도메인에 대한 지식과 프로세스 마이닝 경험이 있는 전문가가 꼭 참여하여야 하며 분석가의 개인적인 역량에 따라 다른 결과가 나오고

분석 기간이 오래 걸린다는 문제점은 여전히 남아 있다. 다시 말해서 프로세스 마이닝은 프로세스 모델을 자동으로 생성하고 이슈를 찾아내는 여러 도구를 제공하여 현행 프로세스 분석 문제에 대한 해결책은 어느 정도 제시하였으나 이슈에 대한 원인분석 및 목표 프로세스 설계에 대해서는 여전히 작업자 개인의 역량에 의존하는 한계가 남아 있다.

[표 2-3]에서 제시한 5가지 분석 방법은 다양한 프로세스 마이닝 프레임워크를 통해 구현된다. 글로벌 컨설팅 그룹인 KPMG<sup>6)</sup>는 프로세스 마이닝 프로젝트를 수행하기 위한 독자적인 방법론을 보유하고 있다 (Blevi, Delporte, & Robbrecht, 2017). 방법론은 [그림 2-10]과 같이 As-Is와 To-Be 설계단계로 나누어 단계별 수행할 7개의 절차를 설명하고 있다.



[그림 2-10] KPMG의 프로세스 마이닝 수행 방법론  
[Fig. 2-10] KPMG's Method of Process Mining

특히 프로세스 마이닝 결과를 분석할 때는 고객 경험 방법론 (KPMG's customer experience methodology)에 따라 수행한다고 밝히고 있는데, 그 내용은 [표 2-4]와 같이 6가지 필수 요소와 5개의 관점을 정의하고 있다. 즉 프로세스를 개인화, 통합, 시간과 노력, 기대수준, 해결과 공감이라는 5개의

6) <https://home.kpmg/be/en/home/insights/2017/09/process-mining.html>

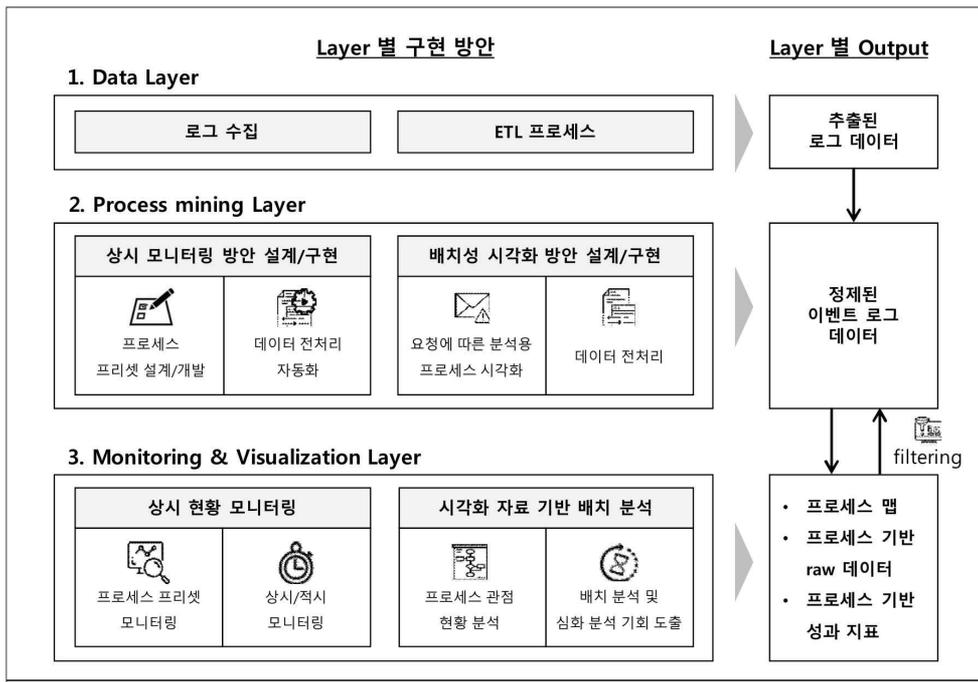
영역으로 나누어, 무엇이 필요한지 정확히 설명하는지, 지연되는 상황을 효과적으로 관리하는지, 대체 솔루션이 제공되는지, 고객에게 문제에 대해 알려주는지, 최신의 효율적인 기술을 사용하는지 등의 5가지 관점 또는 체크리스트를 기반으로 분석한다고 밝히고 있다 (Blevi, Delporte, & Robbrecht, 2017). 이러한 방법론은 프로세스 마이닝을 통해 이슈에 대한 원인을 분석할 때 무엇부터 확인해야 하는지, 어떤 순서로 어떤 항목을 조사해야 할지에 대한 아이디어와 절차를 제공함으로써 프로세스 마이닝에 익숙하지 않은 경우에도 분석할 수 있도록 지원해준다. 그러나 구체적인 알고리즘이 아니라 절차에 불과하므로 이 역시 수행하는 사람의 역량에 좌우되는 것은 마찬가지이며, 직접적인 분석이 아닌 사람이 분석을 도와주는 체크리스트에 가깝다.

[표 2-4] 프로세스 분석 방법론 (Blevi, Delporte, & Robbrecht, 2017)  
 [Table 2-4] KPMG's Customer Experience Methodology

6 Essential Pillars	5 Aspect
<ul style="list-style-type: none"> <li>• Personalisation</li> <li>• Integrity</li> <li>• Time &amp; Effort</li> <li>• Expectations</li> <li>• Resolution and Empathy</li> </ul>	<ul style="list-style-type: none"> <li>• Explain exactly what is needed</li> <li>• Effectively manage delays</li> <li>• Offer alternative solutions</li> <li>• Inform customers about issues</li> <li>• Use latest efficient technology</li> </ul>

권혁진 (2018)은 프로세스 마이닝 기법을 활용한 모바일 게임 이용자 여정 분석에 관한 연구를 통해 [그림 2-11]과 같은 “모바일 게임 이용자 여정 모니터링 및 시각화 기반 분석을 위한 프레임워크”를 제시하였다. “프레임워크의 첫 번째 Data Layer에서는 이벤트 로그의 추출, 변형을 위해 수집된 게임 로그를 ETL 프로세스를 거쳐 데이터웨어하우스에 로그 데이터로 저장하는 과정을 설명하고, 두 번째 Process Mining Layer에서는 프로세스의 특징 및 업무 활용도에 따라 상시 모니터링을 위한 방안과 배치성 시각화를 위한 방안을 제시하며, 세 번째 Monitoring & Visualization Layer에서는 최종 사용자인 실무자 관점에서 일반적으로 본 모니터링 및 시각화 기반의 분석을 위해 수행할 업무 등에 대해 제시”하였다 (권혁진, 2018). 그러나 이러한 프

프레임워크 역시 데이터 수집부터 모니터링, 시각화 절차 및 시각화를 하는 방법을 정의한 것으로 최종적으로 해당 프로세스의 이슈 (여기서는 게임 이용자의 이탈 등)에 대한 원인을 찾아내는 것이 아니라, 사용자의 활동 (Customer Journey)을 프로세스로 표현해주는 방법과 이를 실시간 모니터링하는 방법을 정의한 것이다. 결과적으로 앞에서 복잡한 스파게티 프로세스를 분석하기 위한 다양한 필터와 관련 기능들을 전문가의 직관에 기반하여 적용해보고 이슈에 대한 원인을 유추하는 방식 자체는 같다. 실제로 프레임워크에 따라 시각화된 결과는 앞에서 설명한 프로세스 마이닝에서 생성된 프로세스 모델과 동일하다.



[그림 2-11] 프로세스 마이닝 기법을 활용한 모니터링 및 시각화 구현을 위한 프레임워크의 설계 (권혁진, 2018)

[Fig. 2-11] Process Mining Monitoring & Visualization Framework

결론적으로 프로세스 마이닝을 기반으로 분석을 위한 다양한 분석 방법론 또는 프레임워크들이 연구됐으나 모두 절차, 관점, 체크리스트 등으로 구성되어 있어 본 연구에서 제안된 것처럼 직접적인 결과를 정량적으로 제시하는 알고리즘과는 다른 성격의 연구라고 할 수 있다. [표 2-5]와 같이 기존의 방법론과 프레임워크는 절차와 체크리스트를 제공하며 프로세스 모델이나 그래프 등으로 시각화하는 방법을 통해 프로세스 마이닝을 많이 경험해보지 못한 사람도 어느 정도의 성과를 낼 수 있도록 도와주고 전문가의 실수를 줄여주는 역할을 한다. 그러나 본 연구는 알고리즘만으로 이슈의 원인을 직접 제시하므로 사람의 참여 자체가 없이 것에 의미가 있다.

[표 2-5] 기존 방법론과의 차이  
 [Table 2-5] Differences From Existing Methodologies

	Traditional Method	Proposed Method
Type	<ul style="list-style-type: none"> <li>• Procedure</li> <li>• Checklist</li> </ul>	<ul style="list-style-type: none"> <li>• Algorithms</li> </ul>
Output	<ul style="list-style-type: none"> <li>• Process Model</li> <li>• Graph</li> </ul>	<ul style="list-style-type: none"> <li>• Cause</li> </ul>
Performer	<ul style="list-style-type: none"> <li>• Expert</li> </ul>	<ul style="list-style-type: none"> <li>• None</li> </ul>

## 제 2 절 프로세스 마이닝과 머신러닝

### 1) 머신러닝 알고리즘과 프로세스

머신러닝 알고리즘은 일반적으로 학습 방식에 따라 “지도학습 (Supervised Learning)과 비지도학습 (Unsupervised Learning) 그리고 강화학습 (Reinforcement Learning) 등으로 분류”한다 (Lison, 2015). 또한, 학습 목적

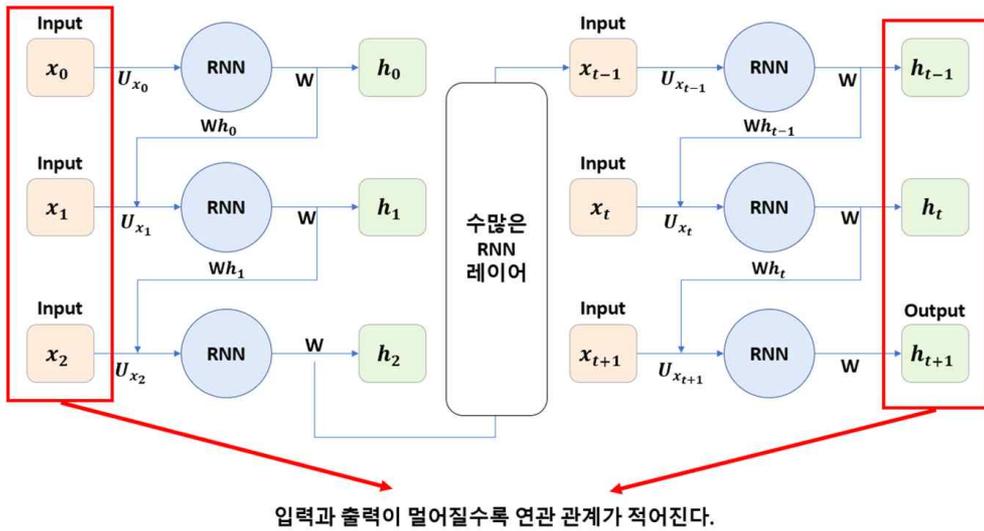
에 따라 분류 (Classification)와 회귀(Regression), 군집 (Clustering) 등으로 분류할 수도 있다.

프로세스 마이닝에서 사용되는 정보시스템 로그는 시간순으로 배열된 선형 (Series) 데이터로 주식 데이터나 자연어 처리를 위한 TEXT 데이터와 유사한 성격을 갖고 있으며, 지도학습을 통해 KPI의 달성 여부를 예측하는 분류 (Classification) 또는 예상 소요 시간 등을 예측하는 회귀 (Regression) 알고리즘이 주로 사용된다.

이러한 선형 데이터의 분류/회귀 예측을 위한 머신러닝 알고리즘에는 자료 집합을 적절한 분할 기준 또는 분할 테스트에 따라 부분 집합들로 나누는 단순 Decision Tree (Rokach, & Maimon, 2005) 알고리즘과 다수의 Decision Tree들을 학습에 사용하는 앙상블 방법인 Random Forest (Breiman, 2001) 알고리즘, 각 특성 사이의 독립을 가정하는 베이지 정리를 적용한 확률 분류기의 일종인 Multinomial Naïve Bayes (Caruana, & Niculescu-Mizil, 2006) 알고리즘, 독립 변수의 선형 결합을 이용하여 사건의 발생 가능성을 예측하는데 사용되는 통계 기법인 Logistic Regression (Cox, 1958) 등이 있다.

최근에는 딥러닝 기반의 선형 알고리즘들이 많이 활용되는데 대표적인 예로 순환신경망 알고리즘인 RNN (Recurrent Neural Network)이 있다. RNN은 유닛 간의 연결이 순환적 구조를 갖는 특징을 갖고 있으며, 시변적, 동적 특징을 모델링 할 수 있도록 신경망 내부에 상태를 저장할 수 있게 해주므로, 일반적인 순방향 신경망과 달리 내부의 메모리를 이용해 시퀀스 형태의 입력을 처리할 수 있다 (Dupond, 2019).

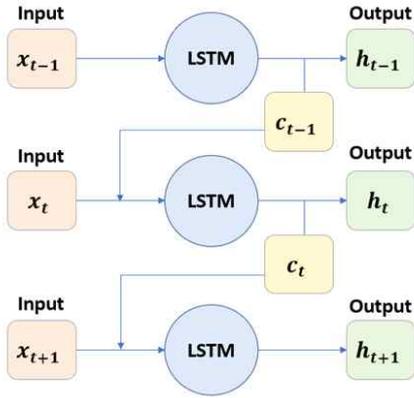
그러나 RNN은 [그림 2-12]와 같이 입력 데이터가 커지면 학습능력이 저하되어 데이터의 뒤쪽으로 갈수록 앞쪽의 입력 데이터를 망각하는 장기 의존성 (Long-Term Dependency)이 발생한다. 그러므로 이를 해결하기 위해 RNN의 변형된 구조인 LSTM (Long Short Term Memory, Gers, Hochreiter, & Schmidhuber, 1997; Schmidhuber, & Cummins, 1999) 또는 GRU (Gated Recurrent Unit, Dey, & Salem, 2017) 등이 주로 사용된다.



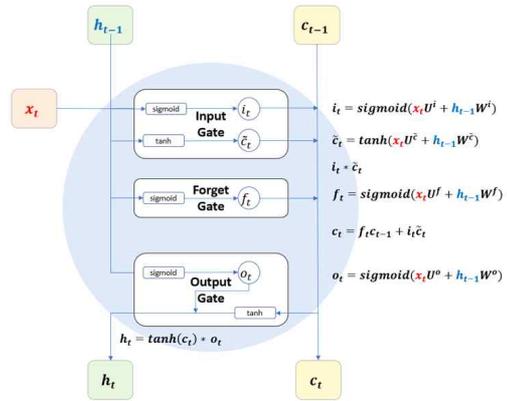
[그림 2-12] RNN의 장기의존성 문제  
 [Fig. 2-12] Long-Term Dependency Issue

LSTM은 [그림 2-13]에 표현된 것처럼 출력값 이외에 LSTM 셀 사이에 공유되는  $c$ 라는 셀 상태 (Cell State) 변수가 추가로 공유된다. 이러한 셀 상태가 다음 레이어로 전달되면서 기존의 상태를 보존하여 장기의존성 문제를 해결할 수 있었다. LSTM 레이어의 각 셀을 좀 더 구체적으로 보면 [그림 2-14]와 같이 Input Gate, Forget Gate, Output Gate라는  $c$  값을 보호하고 제어하는 게이트들이 존재하며 각 게이트는 Sigmoid Layer와 Pointwise 곱 연산을 수행한다 (Gers et al., 1999).

이에 반해 GRU는 LSTM에서의 셀 상태 (Cell State) 역할의  $c$ 가 없으며 대신 다음의 출력  $h$ 에서 그 역할을 함께하여 Update Gate, Reset Gate의 두 가지만으로 더 간단하게 이루어진다. 이를 통해 기존의 LSTM에서 사용되는 셀 상태 계산 (은닉상태 업데이트)를 줄여 좀 더 효율적인 계산이 가능하다. 즉 정확도를 약간 줄이는 대신 시스템 성능 측면에서 강점을 갖는다.



[그림 2-13] LSTM의 구조  
[Fig. 2-13] Structure of LSTM



[그림 2-14]. Cell State의 구조  
[Fig. 2-14] Structure of Cell State

## 2) 머신러닝 기반의 프로세스 예측

최근 프로세스 마이닝과 머신러닝의 융합을 통해 프로세스 마이닝에서 도출된 프로세스 모델을 기반으로 새로운 프로세스가 진행되는 중간에 해당 프로세스의 진행 상황을 모니터링하여 향후 프로세스의 수행 결과를 사전에 예측하려는 연구들이 진행되고 있다.

Van Der Aalst, Schonenberg, & Song (2011)은 통계적인 기법을 통해 진행 중인 프로세스의 남은 수행 시간을 예측하였으며, Ceci, Lanotte, Fumarola, Cavallo, & Malerba (2014)는 Naive Bayes, SVM (Support Vector Machine) 등 다양한 머신러닝 알고리즘을 기반으로 수행 시간과 마지막 수행 Activity를 예측하였다. 최근에는 딥러닝의 종류인 LSTM 알고리즘 기반으로 마지막 프로세스를 예측 (Evermann, Rehse, & Fettke, 2016)을 수행하여 높은 정확도를 보였다.

많은 프로세스 예측 알고리즘 관련 논문에서 표준적으로 사용된 BPI Challenge 2012 데이터 세트 (Van Dongen, 2012)을 기반으로 알고리즘별 정확도를 계산했을 때 [표 2-6]과 같이 Naïve Bayes, SVM 등의 전통적인 머신러닝에 비해 딥러닝 기반의 순환신경망 알고리즘인 LSTM의 정확도가 높았으며 최근에는 89.5% (Theis, & Darabi, 2019)까지 정확도가 향상되었

음을 확인할 수 있다.

[표 2-6] 알고리즘별 프로세스 예측 정확도  
[Table 2-6] Process Prediction Accuracy by Algorithm

Algorithm		Accuracy	Precision	Recall	F1-Score
SVM (Ceci et al., 2014)		0.817	0.856	0.822	0.817
Naïve Bayes (Ceci et al., 2014)		0.617	0.633	0.612	0.555
LSTM	Evermann et al. (2016)	0.824	0.852	0.824	0.817
	Theis, & Darabi (2019)	0.895	0.895	0.895	0.887

※ BPI Challenge 2012 데이터 기준 (Van Dongen, 2012)

또한, 사용된 데이터 세트는 다르지만, 김정연, 윤석준, & 이보경 (2018)은 국내 모 화장품 회사의 프로세스 데이터를 동적 순환신경망 (Dynamic LSTM)을 기반으로 예측해 정확도를 90% 이상까지 높였으며 강영식, 이현우, & 김병수 (2018)는 마지막 수행 Activity 대신 프로세스 수행 결과가 “구매 프로세스의 적기입고”라는 KPI를 만족하는지를 예측할 수 있다는 것을 제시하였다. 실제 기업에서는 단순히 다음 Activity를 예측하는 것보다는 전체적인 프로세스에 대한 KPI를 선정하고 해당 KPI를 달성하지 못하는 프로세스를 사전에 파악하여 선제적으로 적절한 조치를 하는 것이 중요하다는 측면에서 KPI 예측 모델은 기존의 예측 모델보다 기업에 유용할 것으로 판단된다.

### 제 3 절 설명 가능한 인공지능

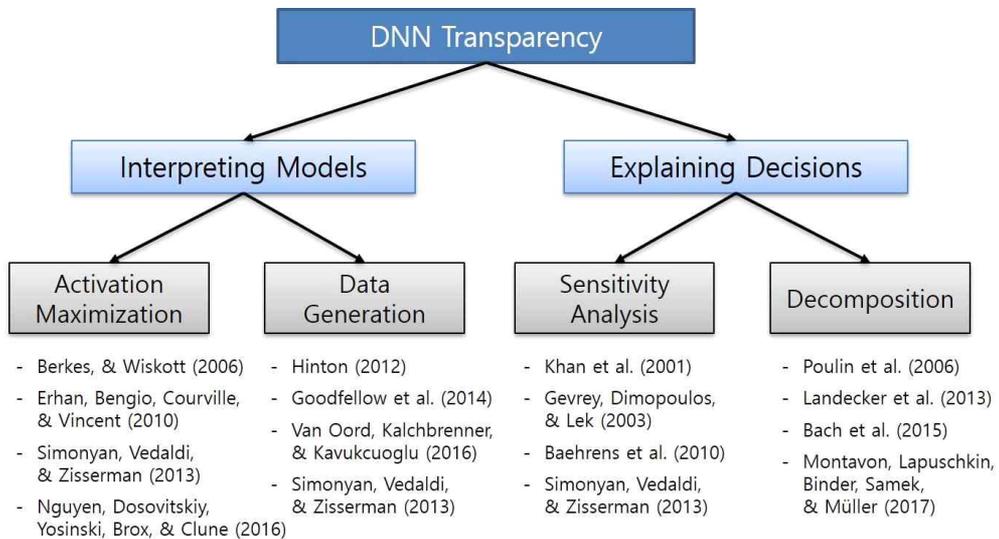
#### 1) 설명 가능한 인공지능의 개념 및 유형

설명 가능한 인공지능 기술은 “인공지능의 개별 의사결정에 대한 설명을 제공하고, 사용자가 인공지능 시스템의 전반적인 강점 및 약점을 이해하도록 도와주기 위해 고안”(Gunning, 2017) 되었으며, 인공지능이 예측한 결과에 대해 주로 영향을 미치는 요소를 찾아내어 예측 결과에 대해 인간에게 직관적인 설명을 도와주는 용도 (Giurgiu, & Schumann, 2019)로도 사용되고 있다.

XAI (eXplainable Artificial Intelligence)를 위한 알고리즘 및 도구들은 현재 DARPA에서 진행 중인 Explainable Learners, Psychology of Explanation 프로젝트 (Gunning, 2017) 외에도 IBM의 AI Fairness 360 (Bellamy et al., 2018), LIME (Local Interpretable Model-agnostic Explanations), LRP (Layer-wise Relevance Propagation, Montavon, Binder, Lapuschkin, Samek, & Müller, 2019) 등 다양한 알고리즘들이 활발히 연구되고 있다.

Montavon, Samek, & Müller (2018)는 [그림 2-15]과 같이 설명 가능한 인공지능 알고리즘을 중요 특징에 따라 크게 “인공지능 모델 자체를 해석하는 방법 (Interpreting Model)”과, “인공지능이 왜 그런 결정을 내렸는지에 대해 데이터로 설명하는 방법 (Explaining Decision)”으로 구분하였다. Interpreting Model은 다시 Berkes, & Wiskott (2006), Erhan, Bengio, Courville, & Vincent (2010), Simonyan, Vedaldi, & Zisserman (2013), Nguyen, Dosovitskiy, Yosinski, Brox, & Clune (2016) 등에서 제시한 Activation Maximization형과 Hinton (2012), Goodfellow et al. (2014), Van Oord, Kalchbrenner, & Kavukcuoglu (2016), Simonyan, Vedaldi, & Zisserman (2013) 등이 주장한 Data Generation형으로 나뉘고 Explaining Decision은 다시 Khan et al. (2001), Gevrey, Dimopoulos, & Lek (2003), Baehrens et al. (2010), Simonyan, Vedaldi, & Zisserman (2013) 등의 Sensitivity Analysis (SA)형과 Poulin et al. (2006), Landecker et al. (2013), Bach et al. (2015), Montavon, Lapuschkin, Binder, Samek, & Müller (2017)의 Decomposition형으로 구분된다. 이때 [그림 2-15]의 가장 왼쪽에 위치한 Activation Maximization이 가장 Model 자체에 집중한다면 (Focus on Model), 가장 오른쪽에 위치한 Decomposition의 경우는 Data에 집중하

여 (Focus on Data) 해석하는 방식이다.



[그림 2-15] XAI 알고리즘 분류 (Montavon et al., 2018)  
 [Fig. 2-15] XAI Algorithm Classification

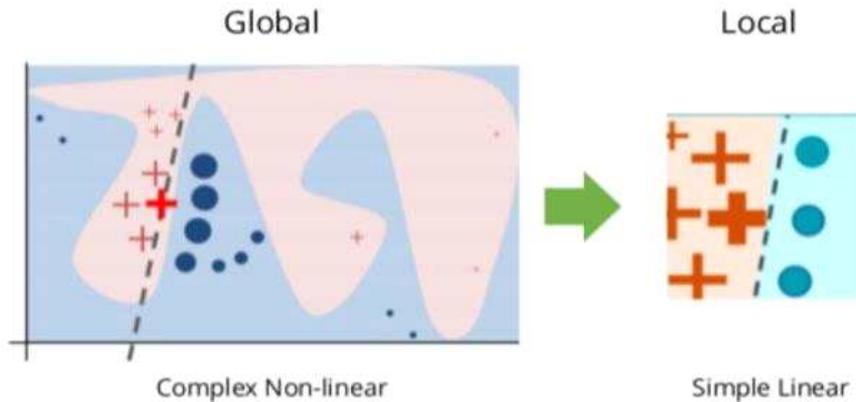
현재 시점에서 많이 사용되고 있는 LIME, SHAP, LRP 등의 알고리즘은 대부분 데이터 중심 (Focus on Data) 유형에 속한다. 그러나 이러한 각각의 XAI 알고리즘은 서로 다른 특성이 있으므로 단순히 알고리즘의 우열을 평가하기보다는 해석의 대상이 되는 데이터와 머신러닝 알고리즘 유형에 따라 적절한 알고리즘을 사용하여야 한다. 그러나 각각의 XAI 알고리즘들에 대한 장단점 분석 및 분야별 활용에 관한 연구는 아직까지 미흡한 상황이다.

## 2) LIME (Local Interpretable Model-agnostic Explanations)

LIME 알고리즘의 핵심 가정은 입력값에 약간의 변화를 주었을 때 모델의 예측 결과값이 크게 바뀌면, 그 변수의 중요도가 높다는 것이다 (Ribeiro, Singh, & Guestrin, 2016). 알고리즘의 명칭에서도 알 수 있듯이 LIME 알고리즘은 로컬해석력을 제공하도록 설계되었다. 즉 예측 모델이 어떤 방식으로

작동하는지를 이해하는 대신 해석하려는 예측값의 근방에서 모델이 어떻게 동작하는지를 설명하며 (Locally Interpretable), 모델에 대해 가정을 하지 않기 때문에 어떠한 모델에도 적용할 수 있다 (Model-agnostic).

해석력의 범위는 글로벌 해석력(Global Interpretability)과 로컬 해석력(Local Interpretability)으로 구분할 수 있다. 글로벌 해석력(Global Interpretability)은 모델에 의해 모형화된 조건분포 전체를 이해할 수 있지만, 대략적이거나 평균적인 값에 근거하게 되는 단점이 있다. 로컬 해석력(Local Interpretability)은 데이터 분포의 작은 영역에 대한 이해를 돕는 데 유용하다. [그림 2-14]와 같이 조건분포의 영역이 작을 때는 선형일 확률이 더 높으므로 Local에 대한 설명이 Global에 대한 설명보다 특정한 결정이나 결과에 대해서 정확도가 더 높을 수 있다.



[그림 2-16] 글로벌 해석력과 로컬해석력 (Ribeiro et al., 2016)  
 [Fig. 2-16] Global Interpretability & Local Interpretability

LIME 알고리즘에서 설명자 (Explainer)는 모델의 종류와 상관없이 로컬 영역에서 설명할 인스턴스의 근처에서 분류기 (Classifier)의 행동을 포착하여 결정경계 (Decision Boundary)를 근사 (Approximation)한다. LIME 알고리즘은 예측 모델의 행동을 분석하기 위해 입력값을 교란 (Perturb)하여 예측이 어떻게 변경되는지 확인한다. LIME의 핵심적인 논리는 통합된 Global Model보다는 단순한 Local Model을 통해 블랙박스인 머신러닝 모델을 근사

하는 것이 더 쉽다는 데에 있다.

이를 수학적으로 표현하면 먼저 설명하고자 하는 사례  $x$ 에서 일부 특성 (Feature)만을 뽑아  $x'$ 을 만든다. 감성 분석에 자주 사용되는 텍스트 문장을 예로 들어보면 "이 영화는 진짜 재미있는 영화다"가  $x$ 라면 '영화'와 '재미'만 사용하는 것이  $x'$ 이 된다. 다음으로  $x'$ 의 근방에서  $z'$ 을 여러 개 뽑아낸다. 예를 들면 '재미'를 빼고 '진짜'를 넣는 방식이다. 그리고  $z'$ 에서 다시  $z$ 를 만든다. 그러면 "영화 진짜 영화"가 될 것이다.

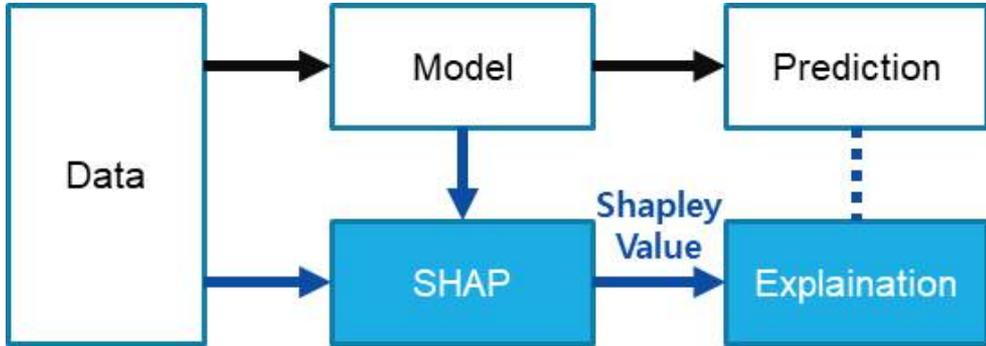
$$\xi(g) = \operatorname{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (2-1)$$

수식 (2-1)과 같이 실제 모형으로 한 예측  $f(z)$ 와 선형 모형처럼 설명 가능한 모형으로 한 예측  $g(z')$ 의 차이( $L$ )와 설명의 복잡성( $\Omega$ )이 가장 작은 (argmin) 모형  $g$ 를 찾으면 이  $g$ 가  $x$ 의 근방에서  $f$ 에 대한 설명이 된다.

### 3) SHAP (SHapley Additive exPlanations)

SHAP 알고리즘은 머신러닝 모델의 출력을 설명하기 위해 게임 이론의 Shapley Values를 기반으로 한 알고리즘이다 (Lundberg, & Lee, 2017). Shapley Values는 협력게임 이론 (Conditional Game Theory)으로 도출된 개념으로 각 Player의 협력과 비협력에 따른 영향을 계산해 Payout을 Align 하는 개념이다. 즉 하나의 특성 (Feature)에 대한 중요도를 알기 위해 여러 가지 특성들의 조합을 구성하여 해당하는 특성의 유무에 따른 평균적인 변화를 측정해서 얻어낸 값이 Shapley Value이다.

[그림 2-17]과 같이 데이터와 학습된 예측 모델을 기반으로 설명을 위한 모델을 생성하고 새로운 입력 데이터가 발생하면 예측 결과에 대한 영향도를 크기와 방향으로 표현한 값인 Shapley Value를 계산하여 입력값이 학습된 모델의 출력값에 대해 어떤 영향도를 가지는지 설명한다.



[그림 2-17] SHAP 알고리즘 작동방식 (정찬일, & 이후진, 2020)  
 [Fig. 2-17] SHAP Algorithm Workflow

SHAP는 Kernel SHAP와 Tree SHAP로 구분된다. Kernel SHAP은 먼저 Feature의 조합을 생성하여 무작위 연합 (Random Coalition)을 생성한 후 각 Feature의 조합을 모델에 적용하여 결과를 얻고 각 Feature 조합들의 가중치 (Weight)를 계산한다. 이때 가중치가 매겨진 데이터를 통해 Linear Model을 만들고 이 모델을 기반으로 Shapley Values를 계산한다. LIME과 Kernel SHAP의 가장 큰 차이는 데이터에 가중치 (Weight)를 매기는 방식에 있다. LIME의 경우 Instance와의 거리를 통해 가까울수록 높은 가중치를 주지만 Kernel SHAP은 Shapley Value의 추정값에 따라 가중치가 달라진다. 각 관측치의 특성에 대한 고유한 영향력을 구할 수가 있다면 모든 특성을 파악할 수 있다. 다시 말해 만약 연합 (Coalition)이 하나의 특성으로만 구성된 경우라면 고유 영향력에 변수 간 상호 작용 더한 값인 총 영향력에 대해 얻을 수 있다. 그러나 연합 (Coalition)의 절반만 특성을 갖소 있는 경우라면 이러한 조합이 무수히 많아지므로 각각의 특성에 대한 영향력을 조금밖에 얻을 수 없다. Lundberg, & Lee (2017)는 수식 (2-2)와 같이 정해진 규칙에 따른 가중치로 Shapley Value를 계산할 수 있다고 제안하였다. 수식에서  $M$ 은 연합 (Coalition)의 최대 크기이고  $|z'|$ 는 관측치  $z'$ 에 존재하는 특성 수이다.

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} (M - |z'|)} \quad (2-2)$$

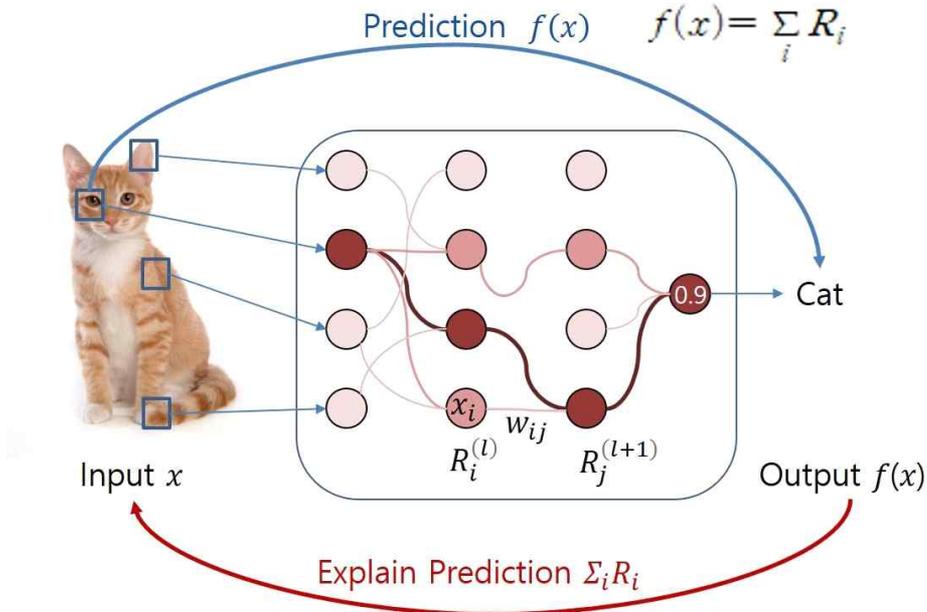
Tree SHAP은 Decision Trees, Random Forests, Gradient Boosted Tree와 같은 Tree 기반의 머신러닝 모델의 해석을 위한 것으로 Kernel SHAP에 비해 분석 속도가 빠르다. SHAP는 특히 Tree 기반 모델에 대해서 빠른 속도를 가지며 LIME과 다르게 Feature Values의 공정한 분배를 통해 예측결과를 평가하고, Global Model Interpretation을 이용하나 Tree 기반 모델이 아닌 경우 속도가 느리며, Feature 간의 종속성을 무시한다는 문제도 있다 (Lundberg, Erion, & Lee, 2018).

#### 4) SA (Sensitivity Analysis)와 LRP (Layer-wise Relevance Propagation)

SA는 LIME과 유사하게 데이터를 중심으로 지역변수의 민감도를 계산하는 알고리즘이고, LRP는 딥러닝 모델에서 각각의 뉴런에 대한 가중치 (weight)를 계산하는 해석 방법이다. SA 알고리즘은 각각의 입력변수들이 모델의 예측 결과에 대해 미치는 영향력의 순위 상관계수를 계산하여 특성에 대한 민감도를 분석한다. 상관계수는 결과(예측변수)와 입력(가정변수) 간의 상관관계의 정도를 나타내는 수치로 만약 가정변수와 예측변수가 높은 상관계수를 갖는다면 가정변수가 예측변수에 대해 유의한 영향력을 갖는다는 것을 의미한다 (Samek, Wiegand, & Müller, 2017).

LRP는 이름 그대로 딥러닝 모델에서 레이어별 (Layer-wise)로 예측 결과에 영향을 주는 관련성 (Relevance)을 구하는 역전파 (Propagation) 알고리즘으로 예측 결과에 대한 딥러닝 신경망의 각 레이어별 기여도를 측정하여, 각 레이어의 결과에 대한 기여도를 히트 맵 형태로 시각화한다 (Bach et al., 2015). 또한, 출력에서 입력 바로 전 단계까지의 모든 뉴런에 대해 활성화 함

수  $a$ 와 가중치  $w$ 의 곱을 역전파 형태로 계산하는 특성 때문에 딥러닝 (Deep Neural Network) 모델에 대해서만 적용할 수 있다.



[그림 2-18] LRP 알고리즘 작동방식 (Montavon et al., 2017)  
 [Fig. 2-18] LRP Algorithm Workflow

LRP는 분해를 통한 설명 (Explanation by Decomposition)을 사용하여 딥러닝 모델을 해석한다. 다시 말하면 임의의 표본  $x = (x_1, x_2, \dots, x_i, \dots, x_d)$ 에 대하여 학습된 딥러닝 모델이  $f(x)$ 라는 출력을 얻는다면,  $d$ 차원 입력  $x$ 의 기여도 (Relevance Score)를 계산하는 것이 목적이다. 이때  $f(x)$ 는 결국 수식 (2-3)과 같이 각 차원  $i$ 에 대한 Relevance Score ( $R_i$ )의 합과 같다 (Bach et al., 2015).

$$f(x) = \sum_i R_i \quad (2-3)$$

예를 들어 [그림 2-18]처럼 특정 딥러닝 모델이 한 장의 사진을 읽은 후

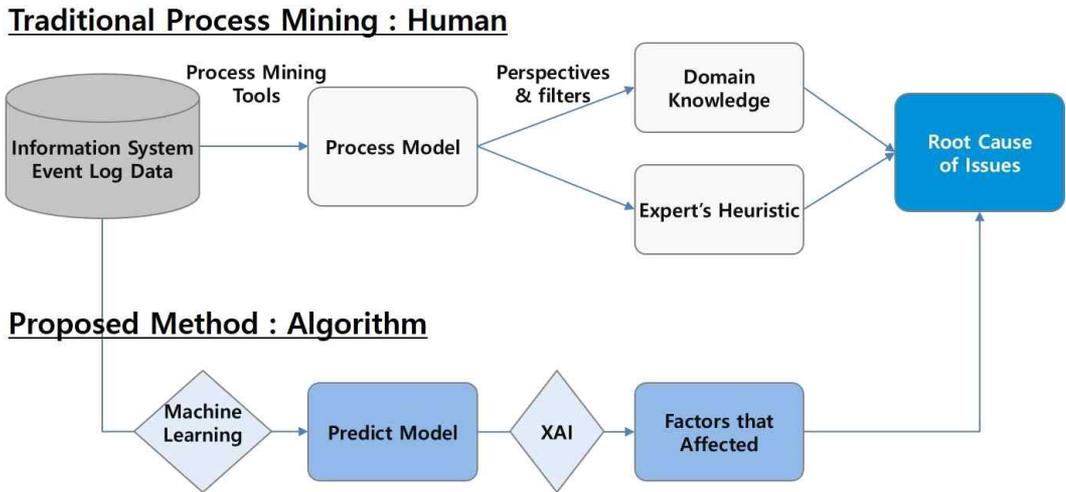
‘고양이’일 확률이 0.99라고 출력했다면 ( $f(x) = 0.99$ ), 해당 모델을 이루고 있는 각 Layer를 구성하고 있는 뉴런 (Neuron)들은 모두 0.99라는 출력에 대한 기여도 (Relevance Score)를 갖고 있으며 Relevance Score의 합 ( $\sum_i R_i$ )은 0.99가 되어야 한다 (Montavon, Lapuschkin, Binder, Samek, & Müller, 2017). 그러므로 기여도를 각 뉴런의 출력단 (Output)에서 입력단 (Input) 방향으로 재분배하여 계산할 수 있으며 그렇게 계산된 결과는 Heat Map의 형태로 표현할 수 있다. 즉 고양이 사진의 귀부분이 기여한 점수, 눈부분이 기여한 점수, 꼬리가 기여한 점수 등을 시각적으로 확인할 수 있다.

# 제 3 장 연구 방법 및 절차

## 제 1 절 가설 및 연구 방법

기존의 전통적인 프로세스 마이닝 프로젝트에서는 정보시스템의 로그 데이터를 기반으로 프로세스 마이닝 도구를 활용하여 자동으로 생성된 프로세스 모델을 분석 관점에 따른 필터를 활용하여 추상화한 후 전문가의 경험과 직관을 통해 문제의 근본 원인을 파악하였다.

본 연구에서는 [그림 3-1]과 같이 정보시스템 로그 데이터를 가공하여 머신러닝 알고리즘을 통해 프로세스의 결과를 예측할 수 있는 모델을 만들고 다시 그 예측 모델을 설명 가능한 인공지능으로 분석하여 예측 결과에 영향을 끼치는 요소 (Activity, Attribute 등)를 자동으로 도출해 낸다. 이슈가 발생될 것으로 예측된 프로세스에 대하여 가장 큰 영향을 끼칠 것으로 예상되는 요소는 결국 전문가가 파악한 이슈 원인과 같을 것이라고 가정하였다. 또한 두가지 요소가 서로 동일하다면 사람의 참여 없이 이슈 원인분석이 가능하다고 가정하였다.



[그림 3-1] 연구 가설  
[Fig. 3-1] Research Hypothesis

예를 들어 [표 3-1]과 같이 5개의 프로세스가 있다고 가정해보자<sup>7)</sup>. ① Process 목록은 정보시스템에서 발생한 이벤트 로그를 각각의 프로세스별로 구분하고 실행시간 순서에 맞게 재조립하여 만든다. 이러한 기능은 프로세스 마이닝 도구에서 기본적으로 제공하며 일반적으로는 Raw 데이터를 그대로 사용하지 않고 사람이 인식할 수 있도록 프로세스 모델 형태로 도식화한다. 이 프로세스를 구성하는 각각의 활동 (A, B, C 등)을 Activity라고 하며 각각의 Activity를 수행한 사람, 수행 시간 등의 속성을 Attribute라고 말한다. 각각의 프로세스들은 일부는 같고 일부는 서로 다른 Activity와 Attribute로 구성된다. (이때 동일한 Activity로만 구성된 Process들을 Case가 같다고 한다.)

[표 3-1] 프로세스 분석 예시  
[Table 3-1] Process Analysis Examples

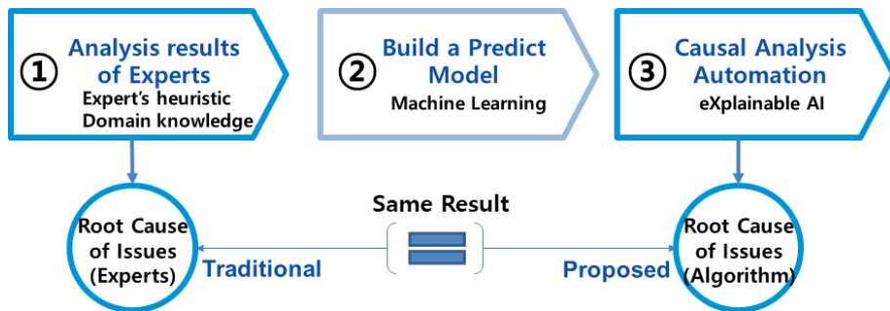
No	① Process	② Result	③ Predict	④ Root Cause
1	A → B → C → D → E	Success (1)	0.78	C = 0.43
2	A → B → F → D → E	Fail (0)	0.12	F = 0.76
3	A → C → D → E → G	Success (1)	0.91	C = 0.32
4	A → B → F → D → E	Fail (0)	0.45	F = 0.51
5	A → B → F → E → G	Fail (0)	0.03	F = 0.42

② Result는 프로세스별로 성과지표의 달성 여부를 별도로 계산하여 부여한다. 예를 들어 적기 배송률이라는 성과지표라고 하면 전체 프로세스의 수행시간을 기준으로 10일 이내면 Success (1)로 10일을 초과하면 Fail (0)으로 라벨링을 한다. 이렇게 ① Process를 전처리하여 수치로 변환한 후 Training

7) 실제 프로세스 마이닝에 사용되는 데이터는 보통 수만에서 수백만 건의 프로세스를 사용한다.

Data로 사용하고, ② Result를 라벨로 사용하는 머신러닝 지도학습 모델을 통해 학습을 진행하면 각각의 프로세스 목록별로 ③ Predict 값을 구할 수 있다. 이 값이 0.5 이하이면 Fail로 예측된 것이고 0.5를 초과하면 Success로 예측한 것으로 판단하여 전체적인 예측 정확도를 계산한다. 그리고 다시 학습된 머신러닝 모델을 XAI 알고리즘으로 분석하면 모델에 입력된 Activity, Attribute 값들이 예측된 결과에 각각 어느 정도의 영향도를 가졌는지 계산할 수 있다. 이때 성과지표를 달성하지 못한 (Fail) 프로세스들 ([표 3-1]에서는 2,4,5번)에 대해서 예측 결과에 가장 많은 영향을 미치는 요소 ([표 3-1]에서는 F)를 이슈에 대한 근본 원인 ④ Root Cause이라고 정의한다.

좀 더 구체적으로 1) A → B → C → D → E 가 각각 “주문 → 포장 → 항공운송 → 입고처리 → 세금계산서발행” 프로세스이고 2) A → B → F → D → E 가 각각 “주문 → 포장 → 해상운송 → 입고처리 → 세금계산서발행”이라면 자동으로 도출된 Root Cause가 “해상운송”이 되므로 해상운송이라는 Activity에 문제가 있다는 의미가 된다.



[그림 3-2] 증명 방법 및 절차 (정찬일, & 이후진, 2020)  
 [Fig. 3-2] Proof Method and Process

이러한 가설을 증명하기 위하여 본 연구에서는 [그림 3-2]과 같이 ① 기존에 프로세스 마이닝 프로젝트를 통해 이슈 원인이 도출된 사례를 찾아 데이터 세트를 입수하고, ② 머신러닝 알고리즘을 이용하여 이슈와 관계된 지표의 결과를 예측한 후, ③ 이 예측 모델을 XAI 알고리즘으로 분석하여 가장 큰 영향을 끼친 항목을 계산하여 그 결과가 실제 프로젝트에서 수행된 전문가들의 분석 결과와 동일한지 확인하였다.

또한, 선행연구에서 알 수 있듯이 프로세스 수행 결과 예측 모델은 Decision Tree, Random Forest, Multinomial Naïve Bayes, Logistic Regression, SVM 등의 전통적인 머신러닝 모델보다 딥러닝 방식의 LSTM 모델의 예측 정확도가 높게 나타난다. 그러나 본 연구의 목적은 원인분석의 정확도에 있으므로 기존의 다양한 머신러닝 알고리즘도 다양하게 적용하여 각각의 알고리즘 간의 장·단점과 원인분석에 미치는 영향을 비교하였다. 특히 설명 가능한 인공지능 알고리즘은 많은 연구가 진행 중인 분야로 알고리즘 간의 비교나 장·단점 분석이 필요한 상황이다. 이를 위해 현재 구현 가능한 소스 코드가 공개된 LIME, SHAP, SA, LRP 알고리즘을 각각의 예측 모델에 적용하여 원인분석의 정확도를 교차 비교하였으며 이를 통해 최소한 프로세스 원인분석 분야에서만큼은 예측 모델과 설명 가능한 인공지능 알고리즘을 선택할 때의 기준이 될 수 있도록 하였다. 또한, 연구 결과의 신뢰도를 높이기 위해 국내외의 다양한 프로세스 마이닝 사례를 조사하고 은행, 제조업, 인터넷 서비스업 등에서 공개된 데이터를 최대한 많이 수집하여 실험함으로써 서로 유사한 결과를 도출하는지 확인하였다.

## 제 2 절 연구 데이터 수집 및 변환

프로세스 마이닝 프로젝트에서 도출된 로그 데이터는 실제 시스템을 통해 수행된 업무의 결과로 대개의 회사에서는 기밀로 취급되고 있어 학술논문에서 활용하기 어려운 측면이 있다. 다행히 네덜란드의 TU/e (Eindhoven University of Technology)에서는 매년 BPI (Business Process Intelligence) Challenge 대회를 통해 유럽의 다양한 기업들에서 개인정보를 제외한 프로세스 데이터를 공개하고 있다<sup>8)</sup>. 이 중 2012년에 공개된 네덜란드의 상업은행인 R은행의 프로세스 로그 데이터 (Van Dongen, 2012)는 프로세스 마이닝과 관련되어 널리 사용되고 있어 주요 연구 데이터로 활용하였다. R은행의 프로세스 데이터는 고객의 대출 신청에서 시작하여 최종적으로 대출이 실행 (입

8) <https://data.4tu.nl/search?q=bpi>

금)되거나 거절 또는 고객의 신청 취소 중의 하나로 끝나는 은행 여신 프로세스 전체에 관한 내용을 담고 있다. 금융 프로세스의 특징은 거의 모든 업무가 정보시스템을 이용하기 때문에 중간에 누락된 업무 활동이 발생할 가능성이 적어 프로세스 마이닝에 유리하다는 데 있다. 예를 들어 고객과 대면 미팅을 하거나 전화 통화를 하는 등의 오프라인 업무가 발생하더라도 은행 규정상 고객과 접촉한 시간과 내용을 모두 시스템에 등록한다. R은행은 이후 2017년에도 다시 한번 BPI Challenge에 데이터를 공개하였으나 비교가 가능한 관련 논문이 많이 있는 2012년 데이터를 사용하였다.

프로세스 마이닝 도구 중 하나인 DISCO의 제조사인 Fluxicon 사에서도 자사의 도구를 활용한 프로세스 마이닝 사례를 일부 가공하여 제공하고 있다. 그중 글로벌 전자제품 제조기업인 P사의 환불 프로세스 일부를 공개한 자료(Rozinat, 2012)를 사용하였다. P사는 네덜란드 소재의 다국적 기업으로 일반 가전뿐 아니라 의료기기와 반도체 분야에서 높은 시장을 점유하고 있는 글로벌 기업이다. 따라서 환불 프로세스도 단순히 대리점과 가전제품 환불이 아닌 국가 간의 물류까지 포함하고 있어 운송 절차 등이 매우 복잡하다.

마지막으로 국내 프로세스 마이닝 전문기업인 피엠아이지<sup>9)</sup>사에서도 국내에서 진행한 여러 프로세스 마이닝 프로젝트의 결과물을 일부 가공하여 공개하고 있는데, 최근에 공개된 음악 스트리밍 서비스 기업인 M사의 모바일 사용자 프로세스 데이터(Loen Entertainment, & PMIG, 2017)도 함께 사용하였다. 일반적인 업무 프로세스와 다르게 웹사이트 또는 모바일 앱의 사용자 로그는 다른 말로 고객 여정 지도(Customer Journey Map)이라고 한다. 이는 표준 업무 매뉴얼에 따라 수행하는 기업 업무 프로세스와 달리 매우 복잡하고 다양한 프로세스 흐름을 보여주며 그 수행 시간도 짧은 대신 웹, 모바일 시스템의 비정형 로그를 통해 추가적인 정보를 확인할 수 있는 특징이 있다. 최근 빅데이터 시스템을 갖춘 기업들이 증가하면서 웹/모바일 앱의 모든 로그 데이터를 저장하게 됨으로써 이를 통한 고객 여정 지도 분석의 중요성이 높아지고 있다. 사용된 3가지 연구 데이터의 특징은 [표 3-2]에 정리하였다.

---

9) <http://www.pmig.co.kr/>

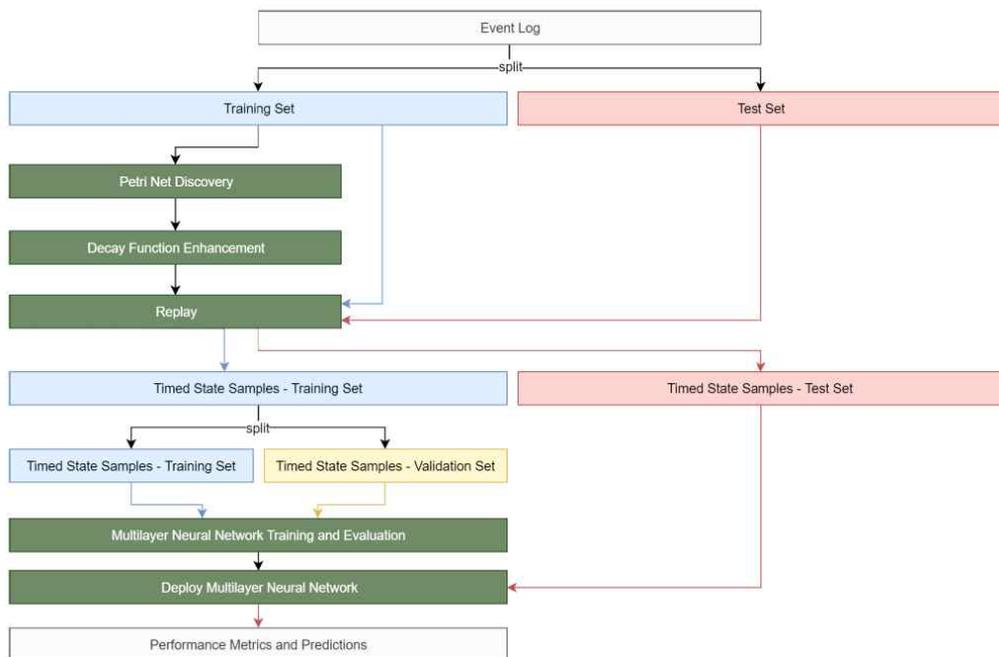
[표 3-2] 사용 데이터 세트 및 포맷  
 [Table 3-2] Used Data Set & Format

Data Set	Country	Industry	Format
Van Dongen (2012)	Netherlands	Bank	.XES
Rozinat (2012)	Netherlands (Global)	Electronic Manufacturing	.DSC
Loen Entertainment, & PMIG (2017)	Korea	Online Service	.DSC

Theis, & Darabi (2019)는 기존에 Evermann, Rehse, & Fettke (2016)이 제시한 프로세스 예측 연구에서 데이터 전처리 방법을 체계화하여 DREAM-NAP (Decay Replay Mining to Predict Next Process Events)이라는 프로세스 예측 방법론 (절차)를 제시하였다. [그림 3-3]과 같이 이벤트 로그 데이터를 Training Set과 Test Set으로 나눈 뒤 로그 데이터의 Timestamp를 기준으로 시간순으로 정렬한 뒤 프로세스 모델의 일종인 Petri Net을 생성하고, 다시 시간에 따른 상태정보를 가진 Timed State Training Set과 Timed State Validation Set으로 분리하였다. 다음으로 데이터 딥러닝 예측 모델인 Multilayer Neural Network에 맞게 가공한 후 학습을 진행하고 최종적으로 Test Set의 데이터를 이용하여 성능을 측정하는 순서이다.

본 연구에서는 Theis, & Darabi (2019)의 DREAM-NAP 절차와 동일하게 수집된 Event Log를 Training Set과 Test Set으로 나누었으며, 프로세스 모델을 확인하였다. 다만 기존 연구에서는 오픈소스인 ProM을 사용해 Petri Net을 생성한데 반해 최근에 많이 사용되는 프로세스 마이닝 도구인 DISCO를 활용하였다. DISCO는 데이터를 시간순으로 정렬하는 기능을 포함하고 있어 이를 이용해 Timed State Training Set과 Timed State Validation Set을 생성할 수 있었다. 또한, Multilayer Neural Network는 기존 연구와 동일하게 LSTM을 사용하였다. Hochreiter, & Schmidhuber (1997)는 장·단기 기억

이 가능한 LSTM 모델을 발표하여 시계열 데이터에 대한 다양한 예측이 가능하도록 하였는데, 정확한 결과값을 얻기 위해서는 모든 입력 변수를 0~1 사이의 수치로 변환하여야만 한다. Evermann, Rehse, & Fettke (2016)의 연구에서는 문자로 표현된 프로세스를 수치로 바꾸기 위해 word embeddings를 사용하였다. 이는 기존에 정의된 사전에 기반하여 각 단어와 수치를 1대1로 매핑하여 변환하는 기법으로 표준 단어사전을 이용한다. 그러나 2개 이상의 단어로 구성된 Activity나 특수문자 등의 문제가 발생하는 경우가 있어 본 연구에서는 프로세스를 구성하고 있는 Activity를 별도의 사전으로 정의하여 숫자로 치환하는 방법을 통해 해결하였다. 또한, 기본적인 LSTM의 구조는 시간과 값의 2차원 형태의 데이터를 입력값으로 받아들이므로 수치화된 각 Activity를 시계열 형태로 변환하여 사용하였으며, Activity 이름 이외의 추가적인 속성 (Feature)을 함께 적용한 경우에는 각 시계열에 여러 개의 수치로 된 속성값을 포함하고 있는 3차원 형태의 데이터를 입력값으로 사용하였다.



[그림 3-3] 데이터 전처리 사례 (Theis, & Darabi, 2019)  
 [Fig. 3-3] Data Preprocessing & Training Approach

### 제 3 절 연구 도구 및 라이브러리

다양한 소스에서 수집된 데이터들은 표준 XML (XES), 프로세스 마이닝 도구 자체 포맷 (DSC 등)으로 서로 다르게 구성되어 있다. 이를 머신러닝과 XAI 알고리즘에 활용하기 위해서는 먼저 기본 데이터 포맷인 CSV (Comma Separated Values) 형태로 변환이 필요하다. XES (eXtensible Event Stream) 포맷은 프로세스 마이닝을 위한 이벤트 로그 데이터의 국제표준 포맷 (IEEE 1849-2016)으로 주로 오픈소스 프로세스 마이닝 도구인 ProM<sup>10)</sup>에서 주로 사용되며 본 연구에서는 파이썬 오픈소스 라이브러리인 PM4PY<sup>11)</sup>를 이용해 XES 데이터를 읽은 후 CSV 포맷으로 변환하였다. DSC는 프로세스 마이닝 도구인 Fluxicon사의 DISCO에서 사용하는 데이터 포맷으로 DISCO의 Extract 기능을 이용해 변환했다. DISCO 프로그램은 Fluxicon사의 홈페이지<sup>12)</sup>에서 시험판을 다운로드가 가능하며 현재는 Windows 버전만 제공된다. 실험에 사용된 프로그램은 2.13.1 Version의 시험판으로 업로드할 수 있는 데이터의 양과 일부 기능에 제약이 있으나 전체적인 프로세스 마이닝 데이터를 확인하고 실험에 사용하는 데에는 충분하다.

파이썬 개발환경은 클라우드 환경으로 가장 적은 비용으로 GPU를 사용할 수 있는 구글의 Colab<sup>13)</sup>을 사용하였다. Colab은 구글에서 커스터마이징한 파이썬 노트북 환경을 제공하는데 파일에 대한 읽기/쓰기를 사용하기 위해서 [그림 3-4]와 같은 코드를 사용하여 구글 드라이브와 연결하였다.

```
from google.colab import drive

drive.mount('/content/drive', force_remount=True)
# 출력되는 URL에 접속하여 verification code 복사 및 붙여넣기
```

[그림 3-4] 구글 Colab 환경 연결  
[Fig. 3-4] Connect to Google Colab

10) <http://www.processmining.org/prom/start>

11) <https://pm4py.fit.fraunhofer.de/>

12) <https://fluxicon.com/disco/>

13) <https://colab.research.google.com/notebooks/intro.ipynb>

대용량의 프로세스 로그 데이터를 변환하기 위해서는 PM4PY를 사용하는 것이 편리하다. PM4PY는 이벤트 로그의 Import/Export, Petri Net의 생성과 Process Discovery, Conformance Checking, Social Network Analysis 등의 프로세스 마이닝 도구에서 제공하는 다양한 알고리즘을 파이썬으로 구현해 놓은 파이썬 라이브러리이다. 실험에 사용된 Version은 2.2.7.2로 파이썬에서는 [그림 3-5]와 같이 PIP를 이용해 쉽게 설치할 수 있다

```
pip install pm4py
```

[그림 3-5] PM4PY 라이브러리 설치  
[Fig. 3-5] Install for PM4PY

설치된 PM4PY에서 `xes_importer`를 Import하면 [그림 3-6]와 같이 XES 파일을 읽을 수 있으며 이를 다시 CSV파일로 저장하였다.

```
from pm4py.objects.log.importer.xes import importer as xes_importer

log_path = "./BPI_Challenge_2012.xes"
parameters = {"timestamp_sort": True}
log=xes_importer.apply(log_path, variant="nonstandard", parameters=parameters)
```

[그림 3-6] XES 포맷 데이터 읽기  
[Fig. 3-6] Read of XES Format Data

수집된 프로세스의 구조를 확인하기 위해 프로세스 마이닝 도구인 DISCO를 주로 활용하여 프로세스 모델을 생성하였으며, 데이터의 통계정보를 시각화하기 위해서는 파이썬 라이브러리인 Matplotlib<sup>14)</sup>를 활용하여 직접 파이썬 코드를 작성하였다. 데이터 전처리를 비롯한 머신러닝 모델 및 XAI 알고리즘을 위한 코드는 모두 파이썬 (Python)<sup>15)</sup> Version 3.8을 이용하여 작

14) <https://matplotlib.org/>

15) <https://www.python.org/>

성하였다. 파이썬 및 기본 라이브러리는 Colab에서 기본으로 설치하여 제공하므로 별도의 설치의 필요 없다.

머신러닝 알고리즘 중 Decision Tree, Random Forest, Multinomial Naïve Bayes, Logistic Regression 등은 [그림 3-7]과 같이 파이썬 라이브러리인 싸이킷런 (scikit-learn)<sup>16)</sup>을 사용하였다.

```
from sklearn import tree #Decision Tree
from sklearn.ensemble import RandomForestClassifier #Random Forest
from sklearn.naive_bayes import MultinomialNB #Multinomial Naïve Bayes
from sklearn.linear_model import LogisticRegression #Logistic Regression
```

[그림 3-7] 머신러닝 라이브러리 사용  
[Fig. 3-7] Use of Machine Learning Library

딥러닝 알고리즘인 LSTM은 Tensorflow<sup>17)</sup> Version 2.3을 사용하여 [그림 3-8]과 같이 Sequential 함수로 개발하였다.

```
from tensorflow import keras
from tensorflow.keras import Input, Model
from tensorflow.keras.layers import LSTM, Bidirectional, Dense, Embedding
from tensorflow.keras.models import Sequential

model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM,
                    input_length=MAX_SEQUENCE_LENGTH-TRIM_CNT))
model.add(Bidirectional(LSTM(174)))
model.add(Dense(1, activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=METRICS)
```

[그림 3-8] 예측 모델 생성  
[Fig. 3-8] Make a Predict Model

16) <https://scikit-learn.org/>

17) <https://www.tensorflow.org/>

XAI 알고리즘 중 LIME은 Ribeiro et al. (2016)의 저자인 Ribeiro가 직접 개발한 LIME 라이브러리<sup>18)</sup>와 ELI5 라이브러리<sup>19)</sup>를 함께 사용하였는데 두 라이브러리의 알고리즘은 같지만 시각화 기능이 서로 달라 병행하여 사용했다. 또한, SHAP 알고리즘은 Lundberg, & Lee (2017)를 발표한 Lundberg가 공개한 SHAP 라이브러리<sup>20)</sup>를 기반으로 작성하였다. LIME, ELI5, SHAP 알고리즘은 PIP 기반 설치환경을 제공하므로 [그림 3-9]와 같이 간단하게 설치할 수 있다.

```

pip install lime
pip install eli5
pip install shap

```

[그림 3-9] XAI 라이브러리 설치  
[Fig. 3-9] Install of XAI Library

그러나 SA와 LRP 알고리즘은 Fraunhofer HHI, TU Berlin, SUTD Singapore의 공동 프로젝트인 [heatmapping.org](http://heatmapping.org)<sup>21)</sup>에서 제공한 LRP for LSTM 코드<sup>22)</sup>와 KAIST, UNIST, 서울대학교, 고려대학교, 연세대학교에서 함께 운영하는 설명 가능한 인공지능 연구센터 (Open XAI)<sup>23)</sup>에서 Pycon 한국 2019<sup>24)</sup>에서 공개한 파이썬 소스 코드<sup>25)</sup>를 기반으로 프로세스 마이닝 데이터에 맞게 재개발했다. 특히 LRP는 각각의 Hidden Layer에 존재하는 퍼셉트론의 가중치를 역전파하면서 계산하여야 하므로 난이도가 높은 코드이다.

마지막으로 논문에서 사용된 모든 전처리된 데이터와 파이썬 소스 코드는 저자의 Github<sup>26)</sup>에 공개하였다.

18) <https://github.com/marcotcr/lime>

19) <https://eli5.readthedocs.io/en/latest/>

20) <https://github.com/slundberg/shap>

21) <http://www.heatmapping.org/>

22) [https://github.com/ArrasL/LRP\\_for\\_LSTM](https://github.com/ArrasL/LRP_for_LSTM)

23) <https://www.openXAI.org/>, <http://xai.kaist.ac.kr/>

24) <https://archive.pycon.kr/2019/>

25) <https://github.com/OpenXAIProject/PyConKorea2019-Tutorials>

## 제 4 장 실험 및 분석 결과

### 제 1 절 은행 여신 프로세스 분석 자동화

#### 1) 금융 데이터 특성 및 전문가 분석 결과

Evermann, Rehse, & Fettke (2016)과 Theis, & Darabi (2019) 등 딥러닝 기반의 프로세스 예측 관련 선행연구에서 공통으로 자주 사용된 8th International Workshop on Business Process Intelligence (BPI) Challenge 2012 데이터 세트 (Van Dongen, 2012)는 네덜란드의 R은행에서 수집된 실제 데이터로 [표 4-1]과 같이 총 13,087건의 Case<sup>27)</sup>와 내 262,200건의 Event<sup>28)</sup>로 구성되어 있으며 평균 수행 기간은 8.62일, 최장 수행 기간은 137일까지 걸린다.

[표 4-1] 사용 데이터 세트 특성 (Van Dongen, 2012)  
[Table 4-1] Used Data Set Feature

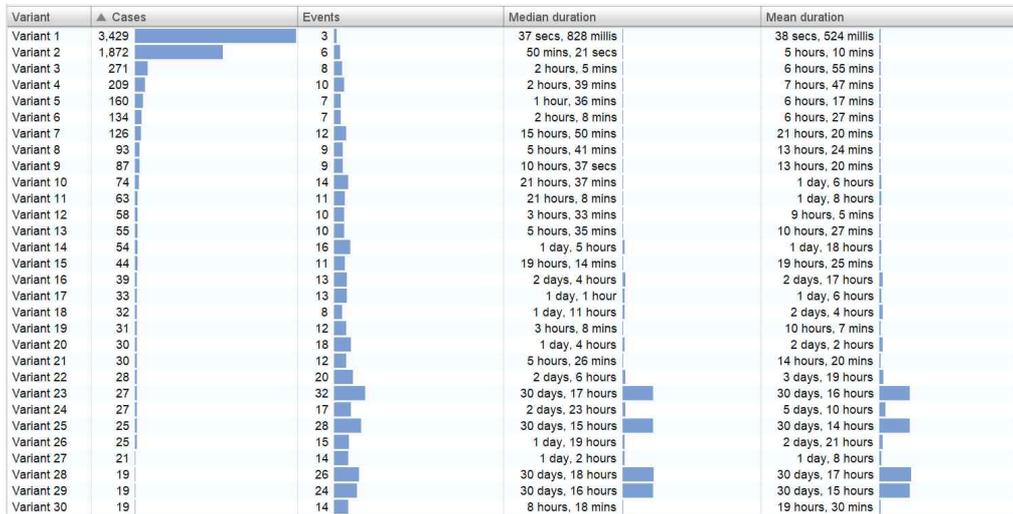
Data Feature	Count
Number of Cases	13,087
Number of Events	262,200
Distinct Activities	36
Distinct Paths	4,366
Average Events per Case	20.03
Average Case Duration	8.62 (Days)
Case Duration (Range)	0 - 137 (Days)

26) [https://github.com/bankplus/process\\_issue\\_analyzer](https://github.com/bankplus/process_issue_analyzer)

27) Case는 여러 개의 Event로 이루어진 프로세스의 집합

28) Event는 시스템에서 행위가 발생한 기록으로 Activity와 동일한 의미임

BPI Challenge의 참가자들은 프로세스 마이닝 기법을 활용한 분석을 통해 7일이 제출에서 승인까지 모든 단계를 수행하는 데 필요한 최소 일수이며 전체 Case의 80% 이상이 23일 이내에 마감되고, 30일 이상부터는 신청 취소 건수가 크게 증가하는 임계점이므로 수익성 향상을 위해서 30일 이내의 프로세스 처리에 추가적인 노력을 들일 필요가 있다'라고 분석하였다 (La Rosa, & Soffer, 2013). DISCO를 통해 분석해본 결과 동일한 Activity가 같은 순서대로 수행된 Case의 집합인 Variant는 [그림 4-1]과 같이 4,366개로 구성되며 3,429개의 Case는 대출 신청 (Application Submitted Complete) → 부분 신청 완료 (Application Partly Submitted Complete) → 승인거절 (Application Declined Complete)의 3단계 Event로 구성되었으며 평균 38초 만에 거절로 끝나는 경우이다.



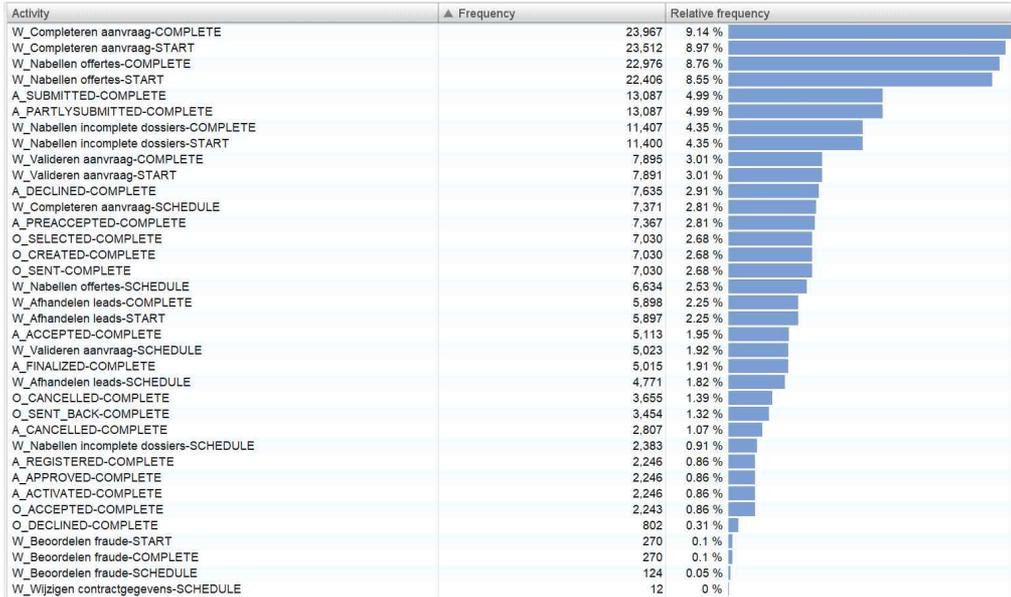
[그림 4-1] 상위 30개 Variant  
[Fig. 4-1] Top 30 Variant

각각의 Case는 36종의 Event의 조합으로 구성된다. Event는 Activity와 동의어로 일반적으로 '한사람이 한자리에서 쉬지 않고 한 번에 처리하는 일'의 단위이다. [그림 4-2]와 같이 이벤트는 영어와 네덜란드어로 구성되어 있고

23,967회 수행되어 전체의 9.14%를 차지한 ‘Completeren aanvraag’은 한국어로 ‘신청 완료’이다. 또한 [표 4-2]와 같이 각각의 Activity 앞에 접두어로 붙어 있는 ‘A\_’는 Application 즉 대출 신청서의 상태를 의미하고, ‘O\_’는 ‘Offer’ 즉 은행의 대출조건 제시 상태, ‘W\_’는 ‘Work Item’ 즉 작업 항목 상태를 의미한다. 또한 각 Activity의 접미어인 ‘COMPLETE’는 완료 상태, ‘SCHEDULE’는 사람의 업무 수행 후에 자동으로 실행되기 위해 대기열에 지정되어 있는 상태, ‘START’는 수행자 또는 시스템이 작업 항목을 배당받은 상태, ‘COMPLETE’는 작업이 끝나거나 다른 작업자의 대기열에 배정(‘SCHEDULE’)된 상태를 의미한다.

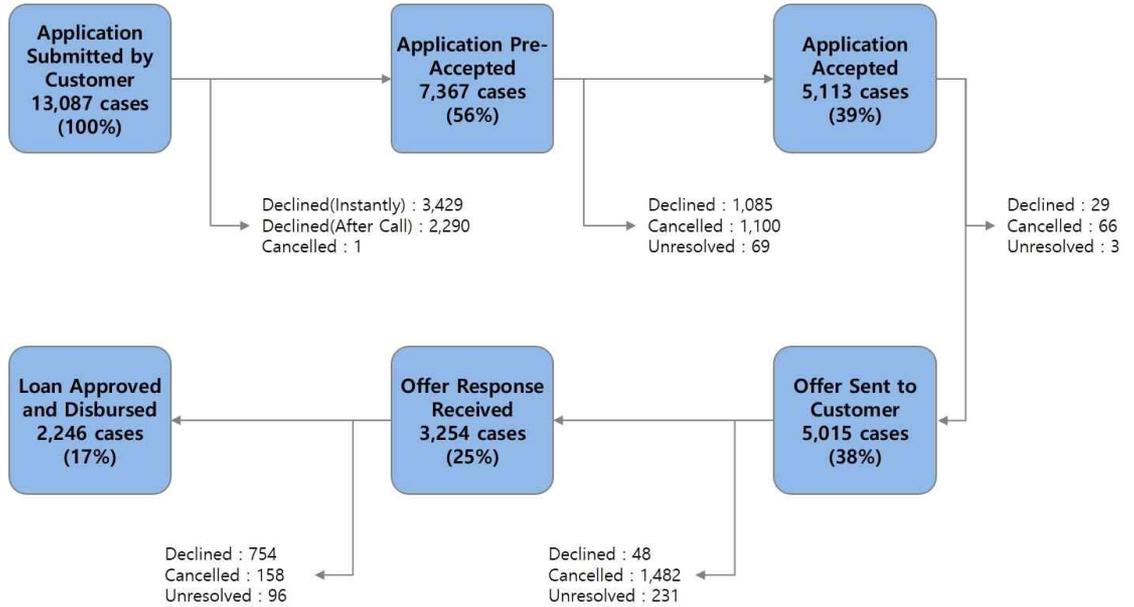
[표 4-2] 이벤트 타입 설명 (Van Dongen, 2012)  
[Table 4-2] Event Type Explanation

Event Type	Meaning
States starting with ‘A_’	States of the application
States starting with ‘O_’	States of the offer belonging to the application
States starting with ‘W_’	States of the work item belonging to the application
COMPLETE	The task (of type ‘A_’ or ‘O_’) is completed
SCHEDULE	The work item (of type ‘W_’) is created in the queue (automatic step following manual actions)
START	The work item (of type ‘W_’) is obtained by the resource
COMPLETE	The work item (of type ‘W_’) is released by the resource and put back in the queue or transferred to another queue (SCHEDULE)

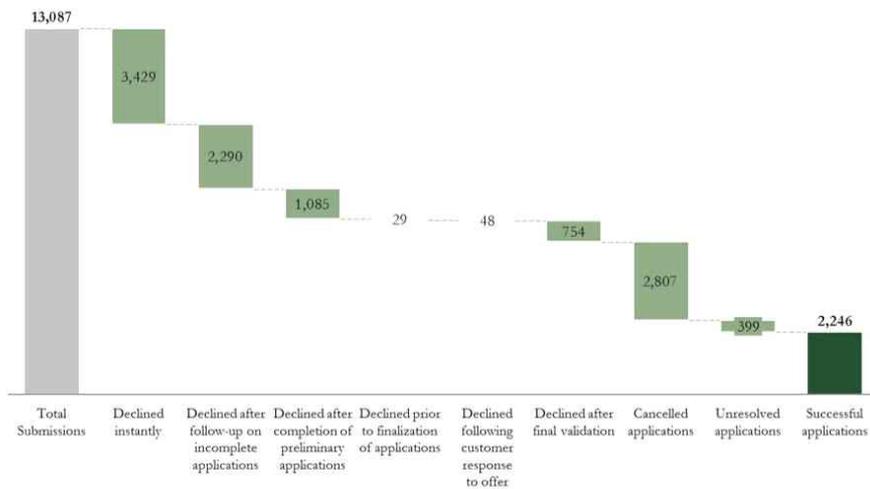


[그림 4-2] Activity 목록 및 실행 건수  
 [Fig. 4-2] Activity List and Frequency

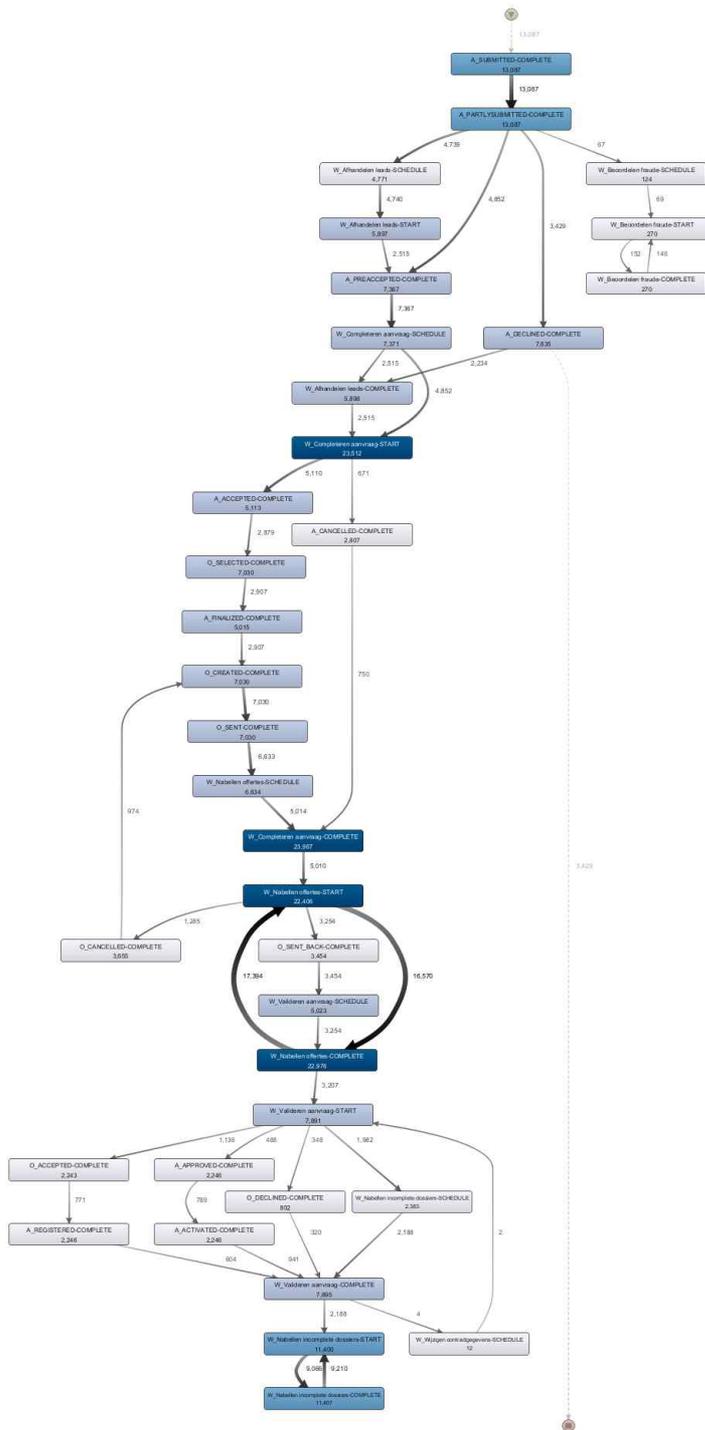
주요한 업무 흐름을 확인해보면 [그림 4-3]과 같이 대출 신청서 제출 (Application Submitted by Customer) 후 대출 신청 사전접수 (Application Pre-Accepted), 대출 신청접수 (Application Accepted), 대출조건 제시 (Offer Sent to Customer), 고객 응답 (Offer Response Received), 최종 승인 및 대출실행 (Loan Approved and Disbursed)의 6개 단계로 구분할 수 있다. 사전접수 시점에 이미 44%의 대출 신청 건이 거부/취소되며, 신청접수까지는 전체의 39%, 고객 응답은 25%, 최종 대출 승인 및 실행까지 완료되는 경우는 전체의 17%에 불과하다. 또한 [그림 4-4]을 보면 고객의 대출 신청접수 후 즉시 거절 (Declined Instantly) 되는 것이 3,429건으로 작성 미 완료된 대출 신청 건에 대한 거절이 2,290건, 고객이 취소하는 경우가 2,807건 등으로 중간에 대출 거절 (Declined), 고객의 취소 (Canceled) 등으로 프로세스가 중단되는 경우가 많으며 최종적으로 대출 실행까지 도달하는 경우는 2,246개의 Case이다.



[그림 4-3] 주요 프로세스 흐름 (La Rosa, & Soffer, 2013)  
[Fig. 4-3] Key Process Steps and Application Volume Flow



[그림 4-4] 신청 건 대비 성공 비율 (La Rosa, & Soffer, 2013)  
[Fig. 4-4] Total Submission vs Successful Application



[그림 4-5] 프로세스 마이닝 모델 (DISCO 활용)  
 [Fig. 4-5] Process Mining Model

BPI Challenge 2012에 참가한 논문들을 정리한 La Rosa, & Soffer (2013)의 내용을 보면, 참가자들은 [그림 4-5]처럼 DISCO를 비롯한 다양한 프로세스 마이닝 도구와 기법을 사용해 분석한 결과 “1) 대출 신청의 약 1/4 (13,087개 중 3,429개)이 즉시 거부되는 것은 신청 접수할 때 자동으로 적용되는 심사 기준이 매우 엄격하다. 2) 나머지 신청 건의 거의 1/4 (9,658개 중 2,290개)가 서류구비 진행 중 거부되어 지속적인 리스크 관리 프로세스가 정상적으로 작동하고 있다. 3) 최종 심사 단계에서 대출 신청의 약 23% (3,254개 중 754개)가 거부되어 사전 단계에서 검토를 강화할 가능성이 있음을 나타낸다.”라고 분석하였다. BPI Challenge의 특성상 전체 프로세스에 대한 분석 중심이므로 이러한 분석 결과를 통해 구체적으로 대출 실행 성공 건수를 늘리기 위해 어떠한 업무혁신이 필요한지에 대한 대안까지는 제공되지 않았다.

은행 여신 프로세스의 수행 결과는 최종적으로 대출 거절 (Declined), 고객의 취소 (Canceled), 그리고 정상적인 대출 실행 (Loan Approved) 중 하나이다. 각 단계별로 다음 Activity와 최종 수행 시간을 예측했던 Evermann, Rehse, & Fettke (2016)과는 다르게 본연구에서는 각 Case 별로 최종 수행 결과에 대해 예측하는 모델을 설계하였다. 특히 최근 심화되고 있는 인터넷 은행의 등장 및 은행 간의 영업 경쟁 등을 고려할 때 대출 신청 건에 대한 고객의 취소 (Canceled)가 증가하고 있으며 이는 여러 금융기관에 대출 신청 후 하나를 고르거나 0.1%의 금리 차이만 나도 금융기관을 옮기는 등 경영환경의 변화에 따른 결과이다. 그러므로 고객의 취소가 발생할 확률이 높은 고객을 사전에 예측할 수 있다면 해당 고객에 대해 금리할인 등의 프로모션을 통해 고객 이탈을 방지하고 또한 프로세스 측면에서 고객 취소가 발생하는 근본 원인을 찾아낸다면 업무 프로세스 개선을 통해 은행의 경영성과를 높일 수 있는 개선 기회를 획득할 수 있을 것이다. 그러므로 본 연구에서는 최종 수행 결과를 고객이 취소 (Canceled)한 경우와 그렇지 않은 경우 (대출 거절 또는 대출실행)의 2가지로 구분하여 데이터를 라벨링하고 고객이 취소하는 경우를 예측하였다.

## 2) 프로세스 데이터 전처리

BPI 2012 홈페이지<sup>29)</sup>에서 다운받은 데이터 세트 파일은 프로세스 마이닝의 데이터 세트 국제표준인 XES<sup>30)</sup> 포맷으로 구성되어 있어, 먼저 PM4PY<sup>31)</sup>의 파이썬 라이브러리를 활용하여 파일을 CSV<sup>32)</sup> 형태로 변환하였다. 해당 파일에서 종료 Activity가 고객 취소인 Canceled (A\_Canceled, O\_Canceled)인 Case만을 분리하였다. 고객 취소로 종료되는 프로세스의 Case는 [표 4-3]에서 알 수 있듯이 전체의 7%인 934건, Event는 전체의 12%인 31,807건이며 그 외 (Declined, Complete)인 경우의 Case는 12,153건, Event는 230,393건이다.

[표 4-3] 고객 취소 프로세스 통계  
[Table 4-3] Statistics of Customer Canceled

Data Feature	Canceled	Others	Total
Number of Cases	934	12,153	13,087
Number of Events	31,807	230,393	262,200
Distinct Activities	29	36	36
Median Case Duration	30.9 day	12.1 hour	19.4 hour
Mean Case Duration	32.4 day	6.8 day	8.6 day

분류된 데이터를 기반으로 Case 별 Activity 명을 시간 순서대로 정렬하여 저장하고 각각의 Activity 명을 단어로 보고 Tokenizer를 이용해 각각의 Activity를 숫자로 인코딩하였다. 인코딩할 때 사용된 Tokenizer는 Activity 목록을 이용해 [표 4-4]의 내용처럼 Dictionary를 생성하였다.

29) <https://www.win.tue.nl/bpi/doku.php?id=2012:challenge>

30) XES Standard, IEEE 1849-2016, 2016.

31) <https://pm4py.fit.fraunhofer.de/>

32) Comma-Separated Values

[표 4-4] Activity 인코딩용 Dictionary  
 [Table 4-4] Dictionary for Encoding

Activity	Index
W_COMPLETERENAANVRAAG-COMplete	1
W_COMPLETERENAANVRAAG-START	2
W_NABELLENOFFERTES-COMplete	3
W_NABELLENOFFERTES-START	4
A_SUBMITTED-COMplete	5
A_PARTLYSUBMITTED-COMplete	6
W_NABELLENIN-COMPLETEDOSSIERs-COMplete	7
W_NABELLENIN-COMPLETEDOSSIERs-START	8
W_VALIDERENAANVRAAG-COMplete	9
W_VALIDERENAANVRAAG-START	10
A_DECLINED-COMplete	11
W_COMPLETERENAANVRAAG-SCHEDULE	12
A_PREACCEPTED-COMplete	13
O_SELECTED-COMplete	14
O_CREATED-COMplete	15
O_SENT-COMplete	16
W_NABELLENOFFERTES-SCHEDULE	17
W_AFHANDELENLEADS-COMplete	18
W_AFHANDELENLEADS-START	19
A_ACCEPTED-COMplete	20

W_VALIDERENAANVRAAG-SCHEDULE	21
A_FINALIZED-COMplete	22
W_AFHANDELENLEADS-SCHEDULE	23
O_Canceled-COMplete	24
O_SENTBACK-COMplete	25
A_Canceled-COMplete	26
W_NABELLENIN-COMPLETEDOSSIERs-SCHEDULE	27
A_REGISTERED-COMplete	28
A_APPROVED-COMplete	29
A_ACTIVATED-COMplete	30
O_ACCEPTED-COMplete	31
O_DECLINED-COMplete	32
W_BEOORDELENFRAUDE-START	33
W_BEOORDELENFRAUDE-COMplete	34
W_BEOORDELENFRAUDE-SCHEDULE	35
W_WIJZIGENCONTRACTGEGEVENS-SCHEDULE	36

생성된 Dictionary를 기반으로 각각의 Activity 순서쌍으로 구성된 Case 들을 [표 4-5]와 같이 Sequence로 변화시켰다. 이때 가장 마지막 Activity가 Canceled로 끝나는 경우를 찾는 것이 목표이므로 모든 Case에서 마지막 Activity를 잘라내고, 마지막 Activity가 O\_Canceled-COMplete 또는 A\_Canceled-COMplete인 경우 Label을 1로 그 외의 경우는 전부 0으로 세팅하였다. 참고로 [표 4-5]와 같이 모든 Case에서 마지막에 위치한 Activity (Last in Case)는 모두 13개로 O\_Canceled이 3,655건, A\_Canceled이 2,807건이다.

[표 4-5] 마지막 Activity 유형  
 [Table 4-5] Last in Case Activity

Activity	Frequency	Relative frequency
W_COMPLETEREN AANVRAAG-COMplete	23,967	9.14 %
W_NABELLENOFFERTES-COMplete	22,976	8.76 %
W_NABELLENOFFERTES-START	22,406	8.55 %
W_NABELLENINCOMPLETEDOSSIERSCOMplete	11,407	4.35 %
W_VALIDERENAANVRAAG-COMplete	7,895	3.01 %
W_VALIDERENAANVRAAG-START	7,891	3.01 %
A_DECLINED-COMplete	7,635	2.91 %
W_AFHANDELENLEADS-COMplete	5,898	2.25 %
O_Canceled-COMplete	3,655	1.39 %
A_Canceled-COMplete	2,807	1.07 %
A_REGISTERED-COMplete	2,246	0.86 %
W_BEOORDELENFRAUDE-COMplete	270	0.1 %
W_WIJZIGENCONTRACTGEGEVENS-SCHEDULE	12	0 %

또한, Case를 구성하는 Activity 중 Canceled와 관련된 Activity는 모두 삭제하였다. 여기서 가장 적은 Activity로 구성된 Case는 3개, 가장 많은 Activity를 보유한 한 (가장 많은 업무 처리 절차가 수행된) Case는 175개의 Activity가 수행되었으며 62일이 걸렸다. LSTM 모델에 적용하기 위해 각각의 Case 별 Activity List의 사이즈를 통일해야 하므로 최대 길이인 175에 맞추어 앞쪽을 0으로 채워 넣는 Padding 작업을 수행하였다. 그리고 마지막 10개 Activity를 삭제하여 전체 프로세스가 종료되기 10단계 이전에 예측할 수 있도록 최종적으로 [표 4-6]처럼 165개의 길이를 갖게 되었다.

[표 4-6] Activity 인코딩 결과 예시  
 [Table 4-6] Sample of Activity Encoding

구분	내용
Activity List	A_SUBMITTED-COMplete → A_PARTLYSUBMITTED-COMplete → W_AFHANDELENLEADS-SCHEDULE → W_AFHANDELENLEADS- START → A_PREACCEPTED-COMplete → W_COMPLETERENAAAN VRAAG-SCHEDULE → W_AFHANDELENLEADS-COMplete → W_C OMPLETERENAAANVRAAG-START → W_COMPLETERENAAANVRAA G-COMplete → W_COMPLETERENAAANVRAAG-START → A_ACCE PTEDCOMplete → O_SELECTEDCOMplete → A_FINALIZEDCOMP LETE → O_CREATEDCOMplete → O_SENTCOMplete → W_NAB ELLENOFFERTES-SCHEDULE → W_COMPLETERENAAANVRAAG-CO Mplete → W_NABELLENOFFERTES-START → W_NABELLENOFF ERTES-COMplete → W_NABELLENOFFERTES-START → W_NAB ELLENOFFERTES-COMplete → W_NABELLENOFFERTES-START → W_NABELLENOFFERTES-COMplete → W_NABELLENOFFERTE S-START → W_NABELLENOFFERTES-COMplete → W_NABELLE NOFFERTES-START → W_NABELLENOFFERTES-COMplete → W_ NABELLENOFFERTES-START → W_NABELLENOFFERTES-COMPLE TE → W_NABELLENOFFERTES-START → W_NABELLENOFFERTES -COMplete → W_NABELLENOFFERTES-START → W_NABELLEN OFFERTES-COMplete → W_NABELLENOFFERTES-START → W_ NABELLENOFFERTES-COMplete → W_NABELLENOFFERTES-STA RT → W_NABELLENOFFERTES-COMplete → W_NABELLENOFFE RTES-START → W_NABELLENOFFERTES-COMplete → W_NABE LLENOFFERTES-START → W_NABELLENOFFERTES-COMplete → W_NABELLENOFFERTES-START → W_NABELLENOFFERTES-COM PLETE → W_NABELLENOFFERTES-START → W_NABELLENOFFER TES-COMplete → W_NABELLENOFFERTES-COMplete → A_Canc eled-COMplete → O_Canceled-COMplete
Encoding & Padding	[ 0 5 6 23 19 13 12 18 2 1 2 20 14 22 15 16 17 1 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4]
Label	1.0

3) 프로세스 결과 예측

전처리된 13,087개의 Case 중 50%를 랜덤하게 분리하고 그중 다시 50%  
 는 테스트용으로 나머지는 검증을 위한 데이터로 아래와 같이 분류하였다. 또

한, 데이터의 분류를 보면 전체 13,087개의 Case 중 7.14%인 934개의 Case만이 1.0 (Canceled)로 비대칭적인 상황이므로 Tensorflow의 `tf.data.experimental.sample_from_datasets`<sup>33)</sup> 기능을 활용하여 데이터 세트의 Weight가 5:5가 될 수 있도록 Label이 1인 데이터를 리샘플링하였다.

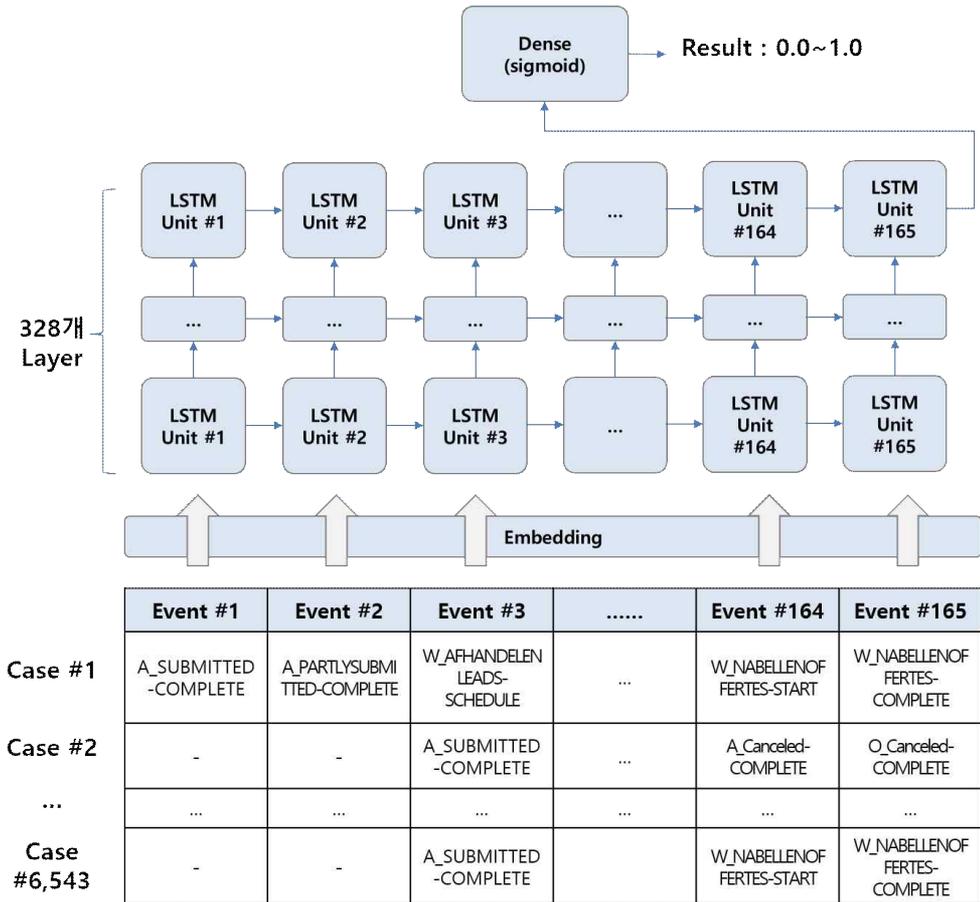
- Train Data : 6,543개 Case
- Validation Data : 3,272개 Case
- Test Data : 3,272개 Case

예측 모델은 [표 4-7]과 같이 Decoding 된 Activity를 다시 Embedding 처리할 수 있도록 Embedding Layer를 추가하고 348개의 Memory Cell을 가진 Bidirectional LSTM Layer를 설정하였으며, 마지막으로 1 (Canceled)과 0 (Not Canceled)를 구분하기 위한 Dense Layer 1개를 추가하였다. 예측 모델의 활성화 함수는 Softmax를 손실함수는 Binary Crossentropy를 사용하였다. [그림 4-6]에서 간략히 정리한 바와 같이 Train Data로 분류된 6,543개의 Case들이 각각의 Event 순서대로 숫자로 변환되어 LSTM의 각 Unit에 입력값으로 들어가게 되고 총 328개의 Bidirectional LSTM 레이어가 중첩되어 최종적으로 Dense 레이어에서 0.0에서 1.0 사이의 결괏값을 예측하게 된다.

[표 4-7] Hyper Parameter (정찬일, & 이후진, 2019b 기반 재작성)  
[Table 4-7] Prediction Model's Hyper Parameters

Layer	Output Shape	Number of Parameter
Embedding	(None, 165, 1)	37
Bidirectional LSTM	(None, 348)	244,992
Dense	(None, 1)	349
Total parameters : 245,378		

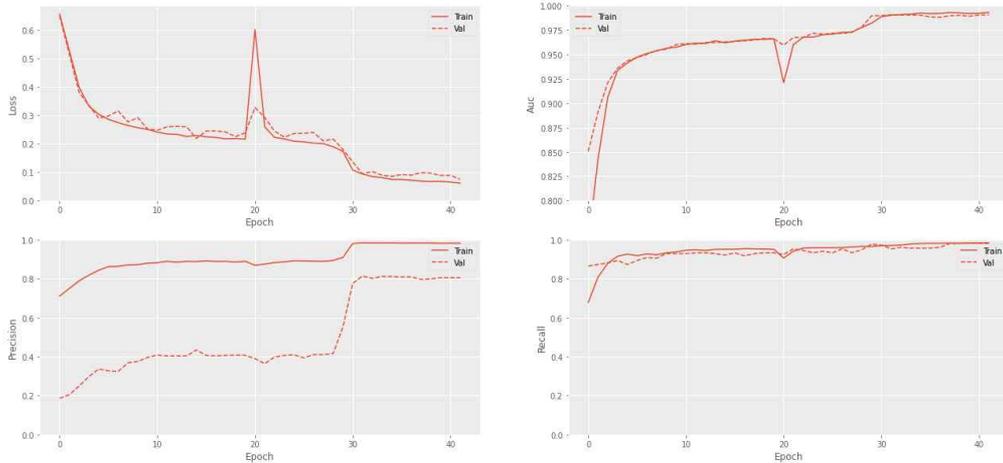
33) [https://www.tensorflow.org/api\\_docs/python/tf/data/experimental/sample\\_from\\_datasets](https://www.tensorflow.org/api_docs/python/tf/data/experimental/sample_from_datasets)



[그림 4-6] 예측 모델의 구조 요약  
[Fig. 4-6] Predict Model's Structure

학습 모델의 과적합을 막기 위한 최적의 학습 횟수를 정하기 위해 Early Stopping 옵션을 AUC (Area Under the Curve)<sup>34)</sup>가 최대로 나올 수 있도록 지정하였으며 AUC가 적더라도 더 학습해보는 Patience 옵션은 10회로 지정하여 학습한 결과 총 50회의 학습을 수행한 후 Stop 되었으며 [그림 4-7]처럼 Accuracy가 0.9639, Validation Accuracy가 0.9704로 학습이 종료되었다.

34) AUC는 모든 임계값에서 분류모델의 성능을 보여주는 ROC (Receiver Operating Characteristics) 곡선 아래 영역의 면적으로 클래스 구분하는 모델의 성능 측정에 사용함



[그림 4-7] 예측 모델 학습 곡선  
 [Fig. 4-6] Predict Model's Accuracy & Loss Learning Curve

이렇게 학습된 예측 모델을 미리 분류해놓은 Test Data Set을 기반으로 모델의 정확도를 테스트한 결과는 [표 4-8]과 같이 Accuracy가 0.9721로 확인되었다. 이는 은행에서 대출 신청이 취소되기 10단계 이전에 97.21%의 확률로 해당 대출 신청 건을 고객이 취소할지 여부를 사전에 예측할 수 있다는 의미이다. 이러한 예측 모델을 실제 은행 여신 시스템에 적용한다면 고객의 신청 취소가 예상되는 대출 건을 사전에 확인하여 금리 우대 등의 다양한 프로모션을 통해 고객 이탈을 방지하는 전략을 활용할 수 있어 은행의 경쟁력 강화에 크게 이바지할 것이다.

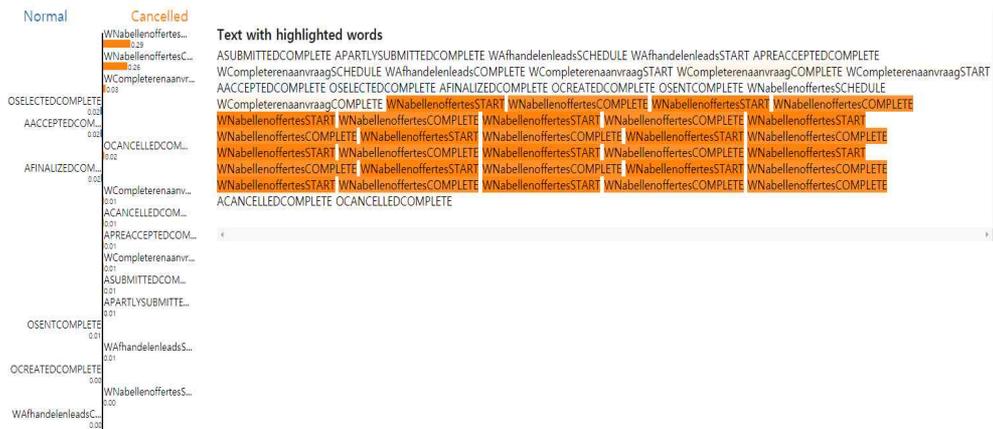
[표 4-8] 예측 모델 정확도 평가  
 [Table 4-8] Prediction Model's Accuracy

구분	Accuracy	Precision	Recall	AUC
Training Data Set	0.9639	0.9675	0.9602	0.9859
Validation Data Set	0.9704	0.7007	0.9431	0.9759
Test Data Set	0.9721	0.7424	0.9406	0.9750

#### 4) 프로세스 원인분석

위에서 작성된 예측 모델을 통해서 고객의 신청 취소 가능성을 사전에 예측할 수 있는 것만으로도 경쟁력 강화에 도움이 되겠지만, 근본적으로 은행의 여신업무 프로세스 중 어떤 부분에서 고객이 떠나가게 하는 문제가 발생하는지를 알아내고, 그 근본 원인을 개선함으로써 업무혁신이 가능하다. 이러한 고객 취소 발생에 대해 여신 프로세스의 문제점이 무엇인지를 알아내기 위해 설명 가능한 인공지능 기법을 적용해보았다.

먼저 LIME 알고리즘을 적용하여 임의의 Case를 분석한 결과 [그림 4-8]와 같이 수행 결과가 'Canceled'일 확률이 0.99로 매우 높았으며 (실제 데이터도 신청 취소 건이었음) 가장 높은 영향을 미친 Activity는 W\_Nabellen offertes (W\_Calling after sent offers, 조건제시 후 고객에게 전화) 시작이 0.2944, 완료 Activity가 0.2628로 기여도가 높았다. LIME에서 제공하는 Text Explainer 기능을 이용해 프로세스 발생 순서를 확인해보면 W\_Nabellen offertes Start와 W\_Nabellen offertes Complete가 14회 이상 반복되며 해당 프로세스 구간이 가장 결과에 많이 기여했음을 색상으로 표시해 준다.



[그림 4-8] 고객 취소 원인분석 결과 #1  
 [Fig. 4-8] Result of Process Issue Analysis #1



## 5) 기존 연구 결과와 비교

BPI Challenge 2012를 통해 대출 신청 프로세스에 대한 분석 결과를 실제 업무에 반영하여 성과를 얻었던 R은행은 2017년도에 대출 신청 프로세스의 업그레이드된 버전을 다시 BPI Challenge 2017<sup>35)</sup>에 제공하였다 (Van Dongen, 2017). 그들은 데이터를 제공하면서 아래의 4가지의 항목에 대해 분석을 요청하였다 (Lopes, & Ferreira, 2019)

- 처리시간, 특히 고객이 은행을 기다리는 시간과 그 반대의 시간
- 대출조건 제안 수락에 대해 영향을 미치는 여러 가지 요소
- 단일 오피 및 다중 오피 고객 간의 비교
- 기타 관심을 가져야 할 트렌드

데이터 제공자의 4가지 요구사항을 중심으로 Student 부문, Professional 부문, Academic 부문별로 총 23개의 팀이 참여하였다. 참가자들은 다양한 기법을 활용하여 데이터를 분석하였지만 대부분 유사한 결과를 제시하였다. 학생 부문에서 수상작은 Povalyaeva, Khamitov, & Fomenko, (2017)으로 이들은 BPMN (Business Porcess Modeling Notation) 다이어그램을 기반으로 적합성을 평가하고, 다양한 도구를 사용하여 성능을 분석하였다. 특히 Random Forrest를 사용하여 프로세스 수행 결과를 예측하였다. 학술 부문의 수상작은 Rodrigues et al (2017)는 DISCO, ProM 등의 다양한 프로세스 마이닝 도구를 사용하였다. 특히 전문가 부문에서 우승한 Blevi, Delporte, & Robbrecht (2017)는 프로세스 마이닝과 글로벌 컨설팅 기업인 KPMG의 고객 경험 방법론을 결합하여 통계를 기반으로 분석 결과를 제시했다. 그러나 KPMG의 방법론은 특정한 기술적 기법이라기 보다는 ① Preparation & Planning, ② Extraction & Transformation, ③ Analysis & Evaluation, ④ Reporting, ⑤ Implement Quick wins, ⑥ Optimization Embedding, ⑦ Continuous Monitoring의 순서로 진행되는 절차에 대한 정의이다. 또한, Logistic

---

35) <https://www.win.tue.nl/bpi/doku.php?id=2017:challenge>

Regression, Random Forrest 및 Deep Learning 모델을 사용하여 프로세스 수행 결과에 대한 예측 모델을 만들었다.

[표 4-9] BPIC에서 사용된 분석 기법 (Lopes, & Ferreira, 2019)  
 [Table 4-9] Techniques Used in the BPI Challenges 2011-2018.

분석 기법	2011	2012	2013	2014	2015	2016	2017	2018	Total
Control-flow discovery	●	●	●	●	●	●	●	●	8
Trace clustering	●	●	●	●		●	●	●	7
Social network analysis	●	●	●	●	●	●	●		7
Performance perspective	●	●	●	●	●		●	●	7
Log statistics		●	●	●	●	●	●	●	7
Conformance checking		●	●	●	●		●	●	6
<b>Predictive modeling</b>				●	●	●	●	●	5
Dotted chart analysis		●	●		●	●	●		5
Plotting/visualization			●	●		●	●	●	5
Trace alignment	●	●	●				●		4
Concept drift analysis					●		●	●	3
Spreadsheet-based analysis		●	●			●			3

전술한 분석 사례에서 알 수 있듯이 효과적인 프로세스 마이닝을 위한 다양한 분석 기법과 알고리즘들이 사용된다. 프로세스 모델에서 업무 흐름을 확인하는 Control-flow discovery, 수행 조직 간의 네트워크 모델을 기반으로 분석하는 Social network analysis, 수행 시간에 대한 프로세스 성능을 중심으

로 확인하는 Performance perspective 등은 분석가 대부분이 즐겨 사용하는 기법으로 초기부터 계속 사용되어 왔다. 그러나 [표 4-9]에서 알 수 있듯이 이전에는 존재하지 않았던, 프로세스 결과 예측 (Predictive Modeling) 기법은 2014년 처음 등장한 이후로는 빠지지 않고 등장하는 중요한 분석 기법이 되었다. 다만 분석 기법이라는 설명이 무색하게 Predictive Modeling을 수행한 모든 연구에서는 최종 결과에 대한 ‘예측 정확도가 00%까지 높아졌다’라는 것으로 결론지어진다. Blevi, Delporte, & Robbrecht (2017)에서도 기존의 다양한 통계 분석 기법으로 이슈에 대한 여러 결론을 낸 이후 별도 섹션으로 분리하여 각각 Logistic Regression, Random Forrest 및 Deep Learning 모델을 사용한 예측 모델의 정확도를 설명하는 것으로 끝나고 있다.

또한 [표 4-10]에서 프로세스 분석을 위한 머신러닝 알고리즘은 큰 분류만으로도 19가지나 되며 초기에는 Decision Tree 등을 통해 Activity 들을 분류, 군집화하는 데 사용되다가 2014년 이후로 프로세스 결과 예측에 사용되기 시작하면서 머신러닝 알고리즘의 종류가 다양해지고 딥러닝 알고리즘도 등장하기 시작한다.

[표 4-10] BPIC에서 사용된 알고리즘 (Lopes, & Ferreira, 2019)  
 [Table 4-10] Algorithms Used in the BPI Challenges 2011-2018.

분석기법	2011	2012	2013	2014	2015	2016	2017	2018	Total
Decision Trees		1		4	2		3	1	12
Logistic Regression				2			2	2	6
Random Forest				1			3	1	5
Neural Network						1	2		3
Linear Regression				2					2
Support Vector Machine				1			1		2

Sequential Pattern Mining				1	1				2
Sequence Classification				1				1	2
Naïve Bayes				1				1	2
Ada Boost				1					1
Apriori Algorithm				1					1
Association Rules				1					1
Multilayer Perceptron				1					1
Binary Segmentation					1				1
K-means Clustering						1			1
Bayesian Networks								1	1
Generalized Linear Model								1	1
Deep Learning								1	1
Gradient Boosted Trees								1	1

이렇게 유사 연구에서 사용했던 분석 기법과 알고리즘을 확인해보면 본 연구와 같이 수행 결과에 대한 예측 중심이 아닌 이슈의 원인분석 중심, 예측 모델이 아닌 설명 가능한 인공지능을 통한 분석 모델을 기반으로 하는 연구는 아직 없는 것으로 판단할 수 있다.

## 6) 실험 결과 및 의미

연구자가 국내의 J은행에서 PI를 수행한 경험에 따르면 최근 인터넷 등을 통한 정보의 공유가 활발해지고 인터넷 은행이 출현하면서, 은행에 대출을 신청한 사람이 다른 은행에서 0.1%만 더 낮은 금리를 제시하여도 대출을 갈아타거나, 아예 대출 신청 시 여러 개 은행에 동시에 신청한 후 가장 유리한 은행을 선택하는 대출상품 쇼핑 현상이 활발해지고 있다. 이는 차주에게 유리한 만큼 은행들의 처지에서는 경쟁이 가속화되는 것을 의미하며 사전에 고객의 성향이나 행동을 예측할 수만 있다면 은행의 영업활동에 큰 도움이 될 수 있을 것이다. 또한, 이런 이슈가 발생하는 프로세스를 사전에 파악하여 제도나 시스템적으로 고객의 신청 취소 건을 줄일 수 있는 방안을 찾는 것은 매우 중요하다.

지금까지 은행의 여신 프로세스 데이터를 기반으로 대출 신청 고객의 취소가 발생할 확률을 딥러닝을 통해 예측해보고, 설명 가능한 인공지능을 통해 예측 결과에 대한 각 Activity의 기여도를 확인함으로써 이슈의 원인을 분석해보았다. 실험 결과를 보았을 때 대출 신청 이후 지속적인 통화가 발생하거나, 신청서의 서류가 반복적으로 수정된다면 고객의 대출 신청 취소가 발생할 확률이 높아진다는 것을 알 수 있었고 이러한 행위 (Activity)들이 시스템상에 발견된다면 사전에 해당 직원에게 알려서 빠르게 조치하거나, 필요할 때 우대 금리를 적용하는 등의 캠페인을 통해 고객의 취소율을 낮출 수 있을 것이라는 결론에 도달하였다. 일반적으로 이러한 업무혁신 또는 개선 아이디어를 얻기 위해서 기존의 PI 프로젝트에서는 업무전문가와 프로세스 전문가가 참여한 조직을 만들고 현업 담당자와의 인터뷰를 거쳐 현황을 분석하고 다시 여러 차례의 회의를 통해 개선 방향을 도출해 낸다. 그러나 프로세스 마이닝과 딥러닝, 설명 가능한 인공지능 기술을 이용하면 특정한 전문가의 참여가 없어도, 여러 단계의 회의와 시간을 들이지 않아도, 단지 업무 프로세스 로그 데이터만 주어진다면 업무 이슈에 대한 원인분석을 자동화할 수 있다는 것을 증명하였다.

## 제 2 절 제조 기업의 환불 프로세스 분석

### 1) 대상 프로세스 특성 및 전문가 분석 결과

글로벌 전자제품 제조사인 P사는 프로세스 마이닝을 통해 환불 처리 (Refund) 업무 프로세스를 개선하였으며, Disco라는 프로세스 마이닝 도구를 개발한 Fluxicon사에서는 그 프로젝트의 결과 및 로그 데이터를 전처리하여 개인정보 등을 제거한 자료를 공개하였다. 해당 프로세스는 영어로 작성되어 있으며 [표 4-11]에서 볼 수 있듯이 1,338건의 환불 처리 Case와 9,032개의 단계 (Event)로 구성되어 있다 (Rozinat, 2012). Event는 Activity가 실행된 사건의 의미로 사용되며 각 Activity에는 해당 업무를 수행한 사람 (Resource)이 명기되어 있으며 이는 숫자로 변환해서 비실명화하였다. 특이한 것은 해당 환불요청이 시작된 3개의 채널 (Dealer, Internet, Callcenter)도 함께 표시하고 있다.

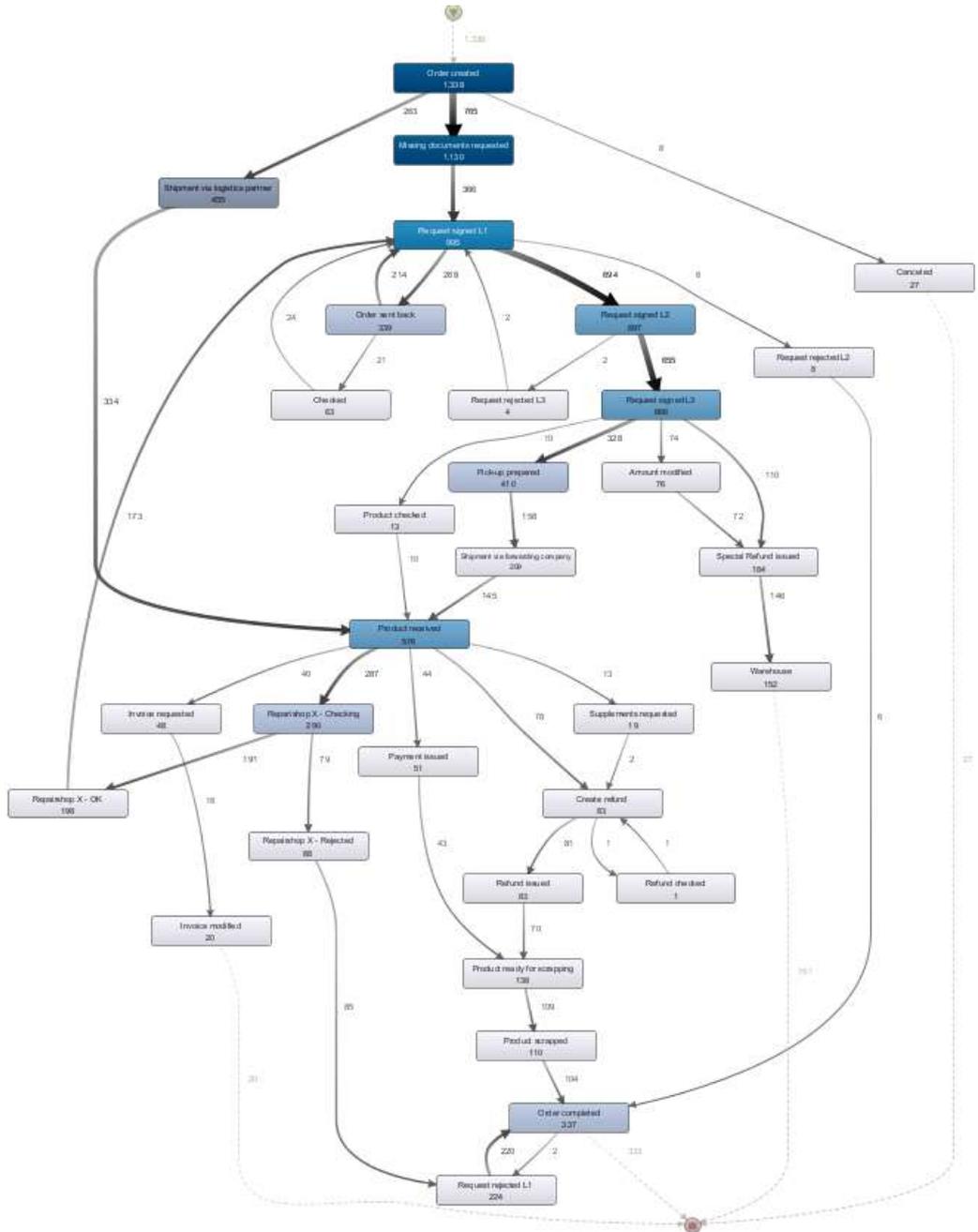
[표 4-11] 환불 프로세스 데이터 세트 특성 (Rozinat, 2012)  
[Table 4-11] Refund Process Data Set Feature

Data Feature	Count
Number of Cases	1,338
Number of Events	9,032
Distinct Activities	32
Median Case Duration (Days)	9 Days
Mean Case Duration (Days)	15.5 Days
Resource (Person)	27 Person
Channel	3 (Dealer, Internet, Callcenter)

총 32개의 Activity는 [표 4-12]과 같이 1,338개의 Case에서 9,032번 등장 (Event)하며, 'Order Created' Activity는 환불업무가 최초 시작되는 역할로 모든 Case에서 100% (1,338번) 발생한다.

[표 4-12] 환불 프로세스 구성 Activity (Rozinat, 2012)  
 [Table 4-12] Activities of Refund Process

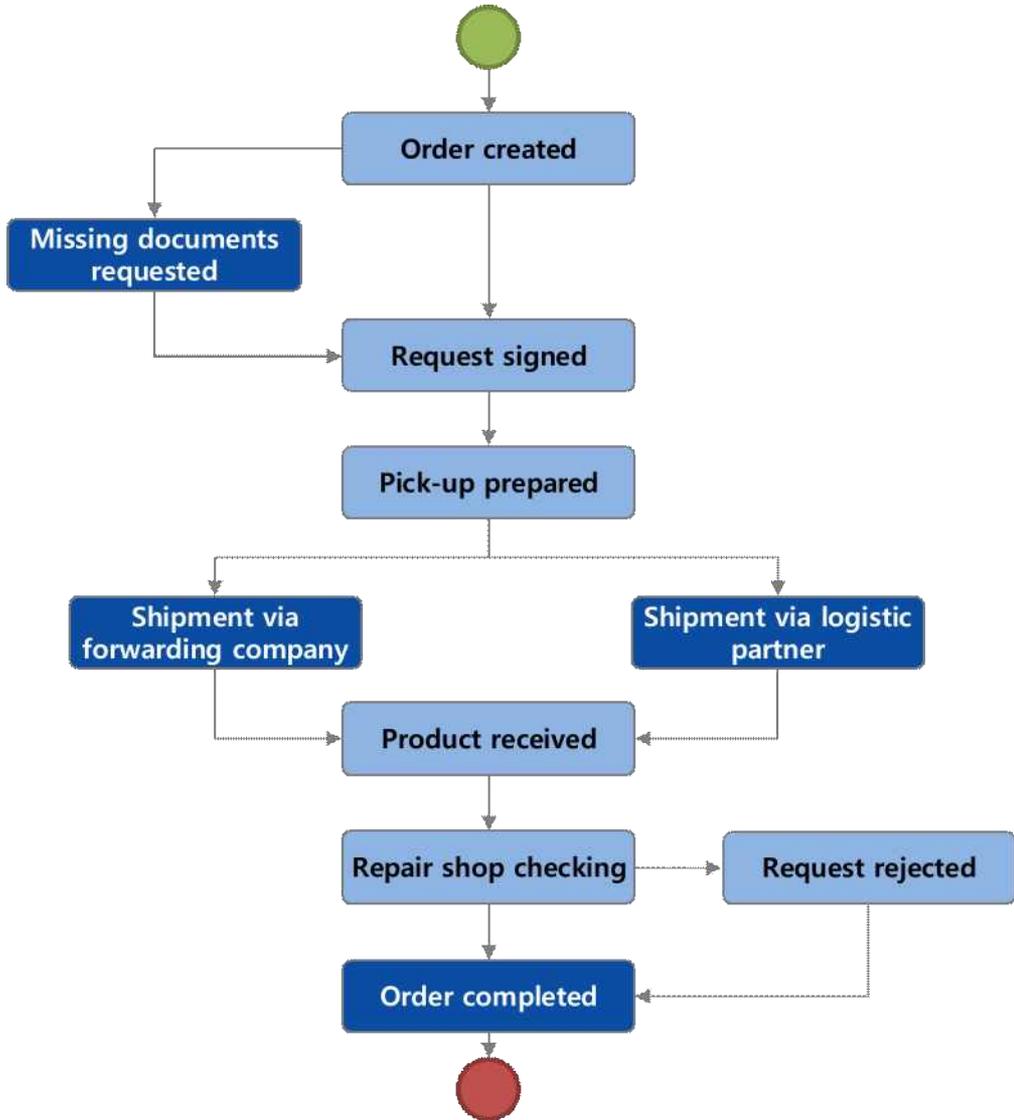
Activity	Frequency	Relative frequency
Order created	1,338	14.81%
Missing documents requested	1,130	12.51%
Request signed L1	995	11.02%
Request signed L2	697	7.72%
Request signed L3	666	7.37%
Product received	576	6.38%
Shipment via logistics partner	455	5.04%
Pick-up prepared	410	4.54%
Order sent back	339	3.75%
Order completed	337	3.73%
Repairshop X - Checking	290	3.21%
Request rejected L1	224	2.48%
Shipment via forwarding company	209	2.31%
Repairshop X - OK	198	2.19%
Special Refund issued	184	2.04%
Warehouse	152	1.68%
Product ready for scrapping	138	1.53%
Product scrapped	110	1.22%
Repairshop X - Rejected	88	0.97%
Create refund	83	0.92%
Refund issued	83	0.92%
Amount modified	76	0.84%
Checked	63	0.70%
Payment issued	51	0.56%
Invoice requested	48	0.53%
Canceled	27	0.30%
Invoice modified	20	0.22%
Supplements requested	19	0.21%
Product checked	13	0.14%
Request rejected L2	8	0.09%
Request rejected L3	4	0.04%
Refund checked	1	0.01%



[그림 4-10] 환불 프로세스 마이닝 모델 (DISCO로 작성)  
 [Fig. 4-10] Refund Process Mining Model

[그림 4-10]는 프로세스 마이닝 도구를 통해 자동으로 생성된 프로세스를 Path 레벨을 낮춰 임의로 추상화한 내용이나 이를 통해 전체 프로세스에 대

해 이해하기에는 아직도 복잡한 수준이다. 이에 전체 환불 프로세스를 최대한 간략화해 보면 [그림 4-11]과 같이 환불 요청이 접수된 후 필요시 누락된 서류를 추가 징구하고 이상이 없으면 계약된 운송주선업자 (Forwarding Company) 또는 물류 파트너사를 통해 제품을 수거하고 수리점에서 점검 후 이상이 없으면 환불이 완료되는 업무 흐름이다.



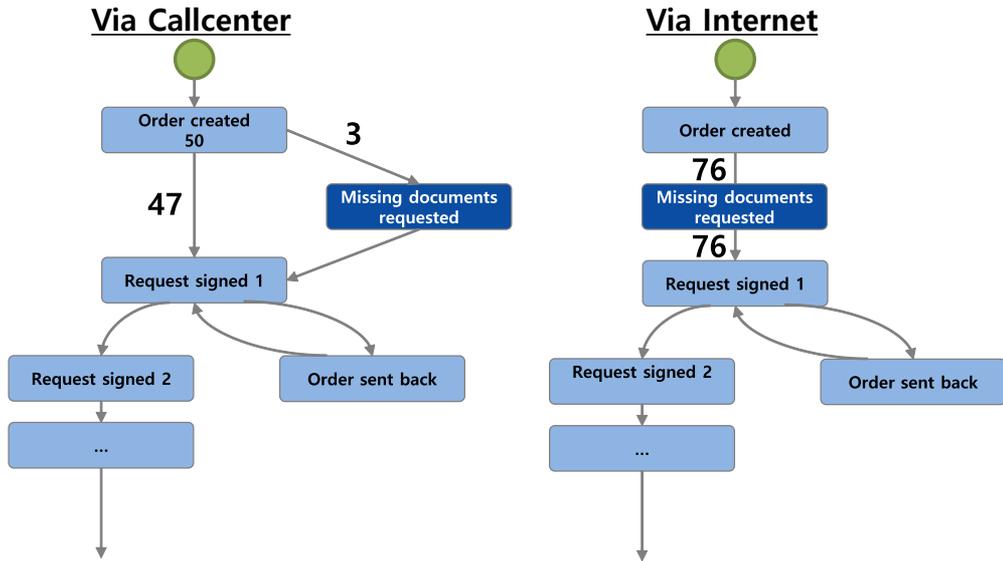
[그림 4-11] 요약된 환불 프로세스 모델  
 [Fig. 4-11] Summarized Refund Process Model

프로젝트에 참여한 전문가들은 환불 프로세스 로그를 기반으로 다양한 프로세스 마이닝 기법과 업무전문가의 노하우를 적용하여 [표 4-13]과 같이 3가지 주요 이슈를 파악하였다.

[표 4-13] 전문가가 분석한 주요 이슈 (Rozinat, 2012에서 재정리)  
[Table 4-13] Issues Analyzed by Experts

Method	Activity	Root Cause	Action
Process Analysis	Missing Documents Requested	Do Not Check Attached Documents When Filing Online	Internet Reception Service Improvement
Performance Analysis	Shipment via Forwarding Company	Delivery Starts Only When the Shipping Pallet is Full	Shipment via Logistics Partner
Conformance Analysis	Payment Issued	Double Deposit, Etc.	Regulation Strengthening and Training

첫 번째 이슈는 프로세스 흐름 분석 (Process Analysis)을 통해 확인된 것으로 인터넷을 통해 환불이 접수된 경우 필수 서류가 누락되어 재요청하는 반복 작업의 발생이다. 콜센터에서 처리된 프로세스와 인터넷을 통해 수행된 프로세스를 [그림 4-12]과 같이 비교해본 결과, 콜센터를 통해 접수된 경우에는 50건 중 3건에 대해서만 누락된 서류에 대한 추가 요청이 발생하였으나 인터넷을 통한 환불 접수시는 76건의 접수건 모두에 대해 ‘추가 서류 요청’ (Missing documents requested)가 발생하였다. 이는 인터넷을 통해 환불 접수 시에는 필수 서류를 체크하지 않았던 것이 원인이었으며, 환불 요청 접수 웹사이트를 개선하여 필수 서류를 자동으로 체크하여 처음부터 고객에게 서류 제출을 요청하도록 개발하였으며 이를 통해 고객 불만을 줄일 수 있었다.

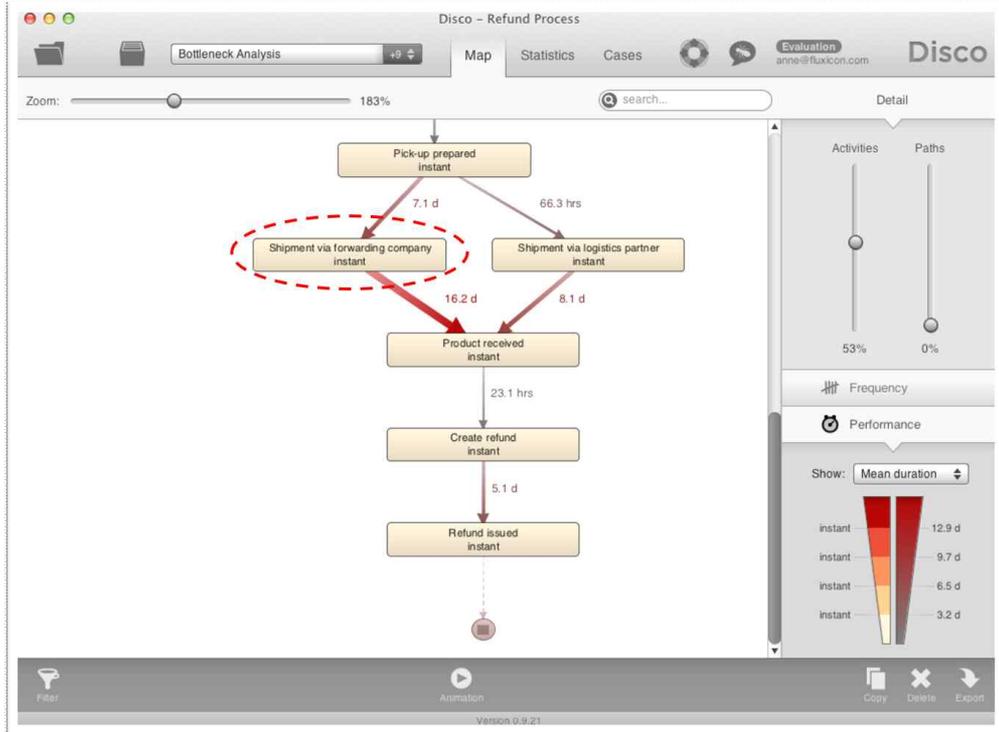


[그림 4-12] 콜센터와 인터넷 환불 프로세스 비교 (Rozinat, 2012)  
 [Fig. 4-12] Call Center vs Internet Refund Process

두 번째 이슈는 업무 프로세스 성능 분석 (Performance Analysis)을 통해 확인된 ‘환불 운송 시간 지연’이다. 환불 요청된 제품을 제휴된 물류회사가 아닌 운송주선업자 (Forwarding Company)를 통해 배송하면 처리시간이 지연되었는데 이것은 운송주선업자가 상품적재용 깔판 (Pallet)<sup>36)</sup>에 화물이 일정량 이상 쌓여야만 배송이 시작하는 것이 원인이었다. 마지막으로 두 번씩 중복 환불하거나 반품된 물건을 확인하지도 않고 환불금을 지급하는 등의 규정 미준수 문제를 발견하였다 (Rozinat, 2012).

이러한 3가지 이슈 중 이슈 원인분석 자동화를 위해 비교적 정량적으로 판단하기 쉽고, 업무적인 영향도도 높은 ‘환불처리 시간 지연’의 근본 원인을 사람의 개입 없이 알고리즘만으로 자동 분석하고 그 분석 결과가 실제 전문가들이 분석한 결과와 같은 ‘운송주선업자를 통한 배송’ (Shipment via Forwarding Company) 문제인지를 확인하였다.

36) 물류센터나 공장 같은 곳에서 사람들이 제품을 차곡차곡 쌓아 놓으면 이후 지게차로 한 번에 들어 차량에 실어 목적지로 보내는 용도로 쓰인다.

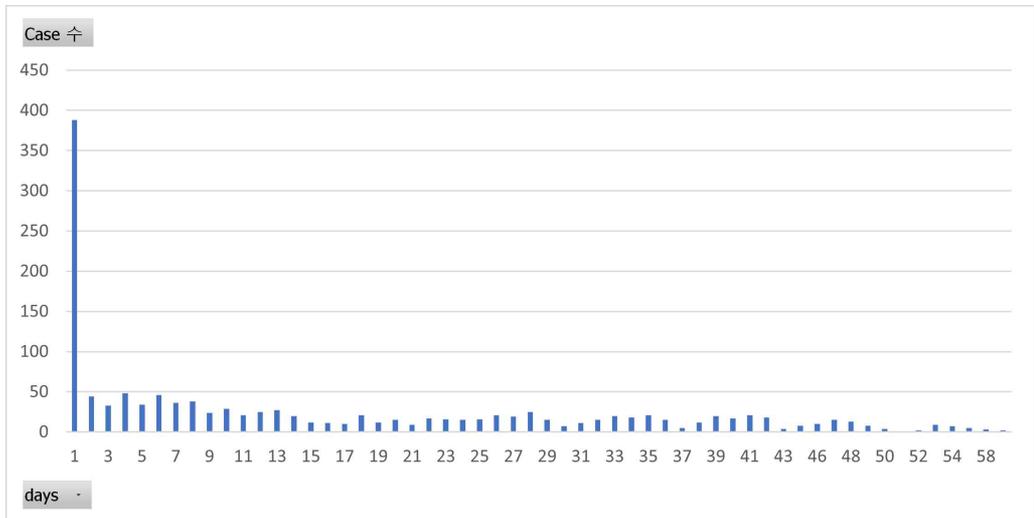


[그림 4-13] Performance Analysis 분석 (Rozinat, 2012)  
 [Fig. 4-13] Performance Analysis

실제 전문가들은 [그림 4-13]과 같이 프로세스 마이닝 도구를 활용하여 각 Event 사이에 수행된 시간을 확인하여 평균적으로 시간이 오래 걸리는 Activity를 찾아낸다. 보통 프로세스 마이닝 도구에서는 시간이 오래 걸리는 Activity의 선 (Line)을 굵기로 표시하거나 색깔을 진하게 하여 시각화해준다. 그러나 예시와는 달리 대부분의 프로세스 모델은 매우 복잡한 스파게티 형태이고 시간이 오래 걸리는 Activity를 발견한다고 해도 해당 Activity가 이슈인지 원래 업무적으로 오래 걸리는 것이 정상인지를 판단하기 어렵고, 해당 Activity 자체에서 시간이 오래 걸리는 것이 아니라 중간에 파악되지 못한 오프라인 작업이 숨어 있어서 지연되는 것인지도 판단해야 한다. 그래서 보통 서로 다른 두 Activity 간의 차이가 명확히 보이도록 하려면 프로세스 마이닝 전문가들이 다양한 Filter를 활용하고 Paths의 Frequency를 여러 차례 조절해가면서 시각화를 진행해야 한다.

## 2) 프로세스 데이터 전처리

사용된 데이터에서 전체 환불 처리 시간(Lead Time)은 중위값은 9일이고 평균값은 15.5일이었으며 [그림 4-13]처럼 1일 이내에 완료되는 Case가 대부분이지만 시간이 오래 걸린 상위 20%의 Case는 30일 이상 걸렸다. 이에 따라 상위 20%의 기준인 30일 이전에 처리가 완료된 경우를 ‘정시처리’ (On Time)로 그 이상 걸린 경우를 ‘지연’ (Delay)으로 분류였다.



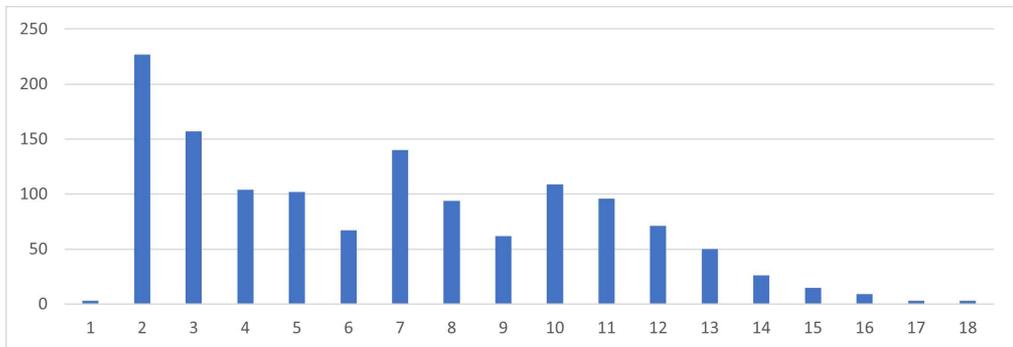
[그림 4-14] Case 별 처리 시간 그래프  
[Fig. 4-14] Duration Chart

[표 4-14]에서 표현된 Variant란 같은 Event 순서를 가진 Case들의 집합이다. 예를 들어 Variant 1은 ‘Order Created’ → ‘Missing Documents Requested’의 2개 Event로만 구성된 Case로 동일한 유형의 Case가 166개나 발생하였고 평균 6분 41초가 걸렸다. 마찬가지로 Variant 2는 3개의 이벤트를 갖고 있으며 69개의 Case들이 동일한 흐름대로 진행되었다는 것을 알 수 있다.

[표 4-14] 상위 10개 Variant  
 [Table 4-14] Top 10 Variant

Variant	Cases	Events	Median duration	Mean duration
Variant 1	166	2	5 mins	6 mins 41 secs
Variant 2	69	3	4 mins	1 hour 50 mins
Variant 3	58	7	7 days 15 hours	8 days 3 hours
Variant 4	57	3	4 days 23 hours	8 days 18 hours
Variant 5	50	10	39 days 47 mins	36 days 12 hours
Variant 6	36	2	6 mins	7 mins 58 secs
Variant 7	35	11	34 days 2 hours	34 days 1 hour
Variant 8	31	5	4 days 2 hours	4 days 2 hours
Variant 9	20	4	2 days 20 hours	5 days 19 hours
Variant 10	19	4	5 hours 35 mins	9 hours 39 mins

또한, 각 Case를 구성하고 있는 Activity들을 [그림 4-15]와 같이 최소 2개에서 최대 18개까지 분포되어 있으며 대부분이 2~10개 사이인 것을 알 수 있다. 예측 모델을 위해서는 Case별 Activity의 갯수를 동일하게 맞추어야 하므로 최댓값인 18개에 맞출 것이다.

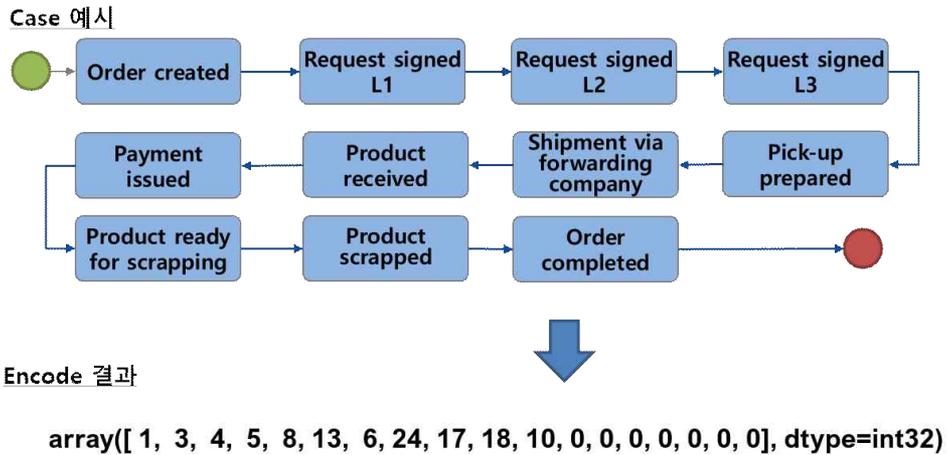


[그림 4-15] Case 별 Activity 개수  
 [Fig. 4-15] Number of Activities per Case

[표 4-15] Activity 수치화 (Word Index)  
 [Table 4-15] Activity Decode (Word Index)

Activity	Word Index	Frequency
Order_created	1	1338
Missing_documents_requested	2	1130
Request_signed_L1	3	995
Request_signed_L2	4	697
Request_signed_L3	5	666
Product_received	6	576
Shipment_via_logistics_partner	7	455
Pickup_prepared	8	410
Order_sent_back	9	339
Order_completed	10	337
Repairshop_X_Checking	11	290
Request_rejected_L1	12	224
Shipment_via_forwarding_company	13	209
Repairshop_X_OK	14	198
Special_Refund_issued	15	184
Warehouse	16	152
Product_ready_for_scrapping	17	138
Product_scrapped	18	110
Repairshop_X_Rejected	19	88
Create_refund	20	83
Refund_issued	21	83
Amount_modified	22	76
Checked	23	63
Payment_issued	24	51
Invoice_requested	25	48
Canceled	26	27
Invoice_modified	27	20
Supplements_requested	28	19
Product_checked	29	13
Request_rejected_L2	30	8
Request_rejected_L3	31	4
Refund_checked	32	1

개별 Activity들은 `tf.keras.preprocessing.text.Tokenizer`<sup>37)</sup> 함수를 사용하여 [표 4-15]와 같이 Word Index를 생성하고 각 Case의 단계를 같게 만들기 위해 18단계보다 짧은 경우는 0으로 채우는 Padding 작업을 수행하였다. 모든 인코딩 작업을 마친 결과는 [그림 4-16]과 같이 수치 배열로 표시하였다.



[그림 4-16] 실제 프로세스 Case 및 인코딩 결과  
[Fig. 4-16] Real Process Case & Encoding Result

### 3) 딥러닝 기반 예측 모델 및 XAI 알고리즘 적용

전처리 완료된 데이터를 기반으로 프로세스 수행 결과를 예측하는 예측 모델을 개발하였다. 예측 모델은 딥러닝 알고리즘 중 프로세스와 같은 시계열 데이터의 결과예측에 적합한 LSTM 알고리즘을 기반으로 [그림 4-17]와 같이 Embedding을 위한 Layer와 128개 Unit을 가진 LSTM Layer, 과적합 방지를 위한 Dropout, ‘정시처리’ (On-Time)와 ‘지연’ (Delay)의 2가지 결과를 예측하기 위한 2개의 Unit을 가진 Dense Layer로 구성하였다. 예측의 정확도를 높이기 위해서는 여러 층의 Layer를 중첩해서 모델을 구현하는 것이 유리

37) [https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/text/Tokenizer](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer)

하나, 추후 XAI 알고리즘을 통해 해당 모델을 분석할 때의 편의를 위해 단층의 LSTM을 사용하였다. 사용된 모델의 활성화 함수는 Softmax를 손실함수는 Binary Cross Entropy를 사용하였다.

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, None, 128)	4224
lstm (LSTM)	(None, 128)	131584
dense (Dense)	(None, 1)	129

Total params: 135,937

[그림 4-17] LSTM 예측 모델의 파라미터  
 [Fig. 4-17] Predict Model's Hyper-Parameter

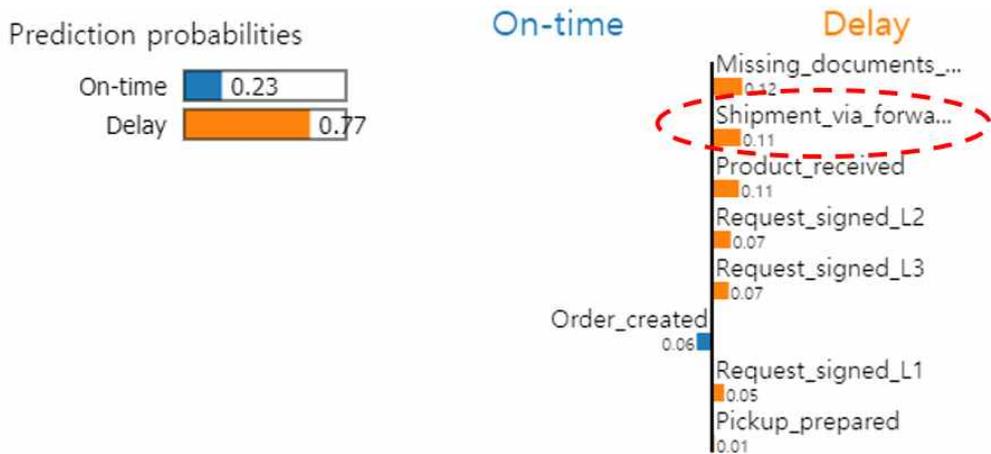
학습한 결과 학습데이터의 Accuracy가 0.8654로 나타났다. 시간 지연 여부를 예측하는 데에 있어 각 Event의 수행 시간을 제거한 채로 단순히 Activity 명과 수행 순서만을 통해 예측하였기 때문에 비록 정확도가 높지는 않게 나타났지만, 프로세스 이슈 원인분석 자동화를 위한 알고리즘을 테스트 하기에는 충분한 성능으로 판단된다. 이후의 실험에서 알 수 있듯이 예측 모델의 정확도를 높이기 위해서는 데이터 전처리를 통해 Activity뿐 아니라 Activity의 경과 시간, 수행 채널(Call Center, Internet), 수행자 (Resource), 고객 불만 접수 여부 등의 추가적인 속성 (Attribute)를 포함시킬 수 있다. 그러나 본 실험의 목적에 따라 다양한 머신러닝 알고리즘과 설명 가능한 인공지능 알고리즘을 적용해보기 위해서는 데이터를 가능한 단순화 시킬 필요가 있었다. 또한, 86%의 예측률은 실제 업무에 적용하기에는 조금 낮은 수준이지만 연구에 필요한 알고리즘 간 비교를 위해서는 충분한 수치라고 판단하였다.



[그림 4-18] LIME 알고리즘 기반 원인분석 #1 (정찬일, & 이후진, 2020)  
 [Fig. 4-18] Sample Process Analysis Using LIME #1

이 예측 모델에 설명 가능한 인공지능 알고리즘인 LIME을 적용하여 예측 결과가 ‘환불처리 시간 지연’ (Delay)인 임의의 Case를 분석한 결과 [그림 4-18]와 같이 전문가가 분석한 지연 원인인 ‘운송주선업자를 통한 배송’ (‘Shipment via Forwarding Company’)가 지연에 영향을 주는 기여도는 3번째로 높게 나타났다. 그러나 기여도가 가장 높은 ‘Order Created’는 음의 기여도를 갖고 있어 예측 결과에 반대의 영향을 준다는 것을 고려한다면 ‘Pickup Prepared’에 이어 2번째로 높은 기여도이다.

이번에는 동일한 LIME 알고리즘을 또 다른 샘플 Case에 적용하여 결과를 확인한 결과 [그림 4-19]과 같이 ‘Delay’일 확률이 77%이며 (실제로도 지연된 Case임) 지연에 영향을 준 Activity는 ‘Missing Documents Requested’에 이어 2번째로 큰 기여도를 보여주었다. 두 개의 샘플 모두 ‘Pickup Prepared’와 ‘Missing Documents Requested’는 양의 영향력을 강하게 주고 있는 것도 확인할 수 있다. 개별적인 샘플 Case를 확인한 결과를 보았을 때는 전체적으로 양호한 결과로 보인다.



[그림 4-19] LIME 알고리즘 기반 프로세스 분석 #2  
 [Fig. 4-19] Sample Process Analysis Using LIME #2

그러나 개별 Case의 영향도를 샘플로 살펴본 것만으로는 정확한 측정이라고 할 수 없다. 이에 프로그램을 통해 전체 1,338건의 Case에 대해 모두 LIME 알고리즘을 적용하여 각 Activity 별로 전체 계산된 기여도를 모두 합산하였다. Activity의 영향도는 개별 Case에 따라 양수이거나 음수인 경우가 모두 있어 굳이 평균을 내는 것보다는 합산을 하는 것만으로도 순위를 확인할 수 있다. [표 4-16]는 전체 Activity에 대해 모든 Case에서의 영향도를 합계한 집계표이다. 모든 경우의 수를 합산했을 때 “Shipment via Forwarding Company”의 기여도 합계가 1.036031였으며 전체 Activity 중 5번째였다. 전체 합계가 -0.14689로 음수인 “Shipment via Logistics Partner”와 대비하면 “Shipment via Forwarding Company”의 영향도 합계는 음수가 아닌 양수이므로 이슈 원인에 대한 방향성은 일치한다고 할 수 있으나 ‘환불 지연’의 주요 원인으로 지목하기는 어려운 수치이다. 물론 상위에 올라 있는 ‘접수 거절’ (Request Rejected), ‘수리점 거절’ (Repairshop Rejected)와 같이 상식적으로 업무가 지연될 것이 자명한 Activity들을 제외한다면 나름대로 의미 있는 수치이지만 본 연구의 목적은 전문가의 개입 없이 결과를 도출하는 것이므로 실험 결과는 만족스럽지 않았다.

[표 4-16] LSTM 모델 Activity별 평균 영향도 (정찬일, & 이후진, 2020)  
 [Table 4-16] LSTM Model Average Delay Effect by Activity

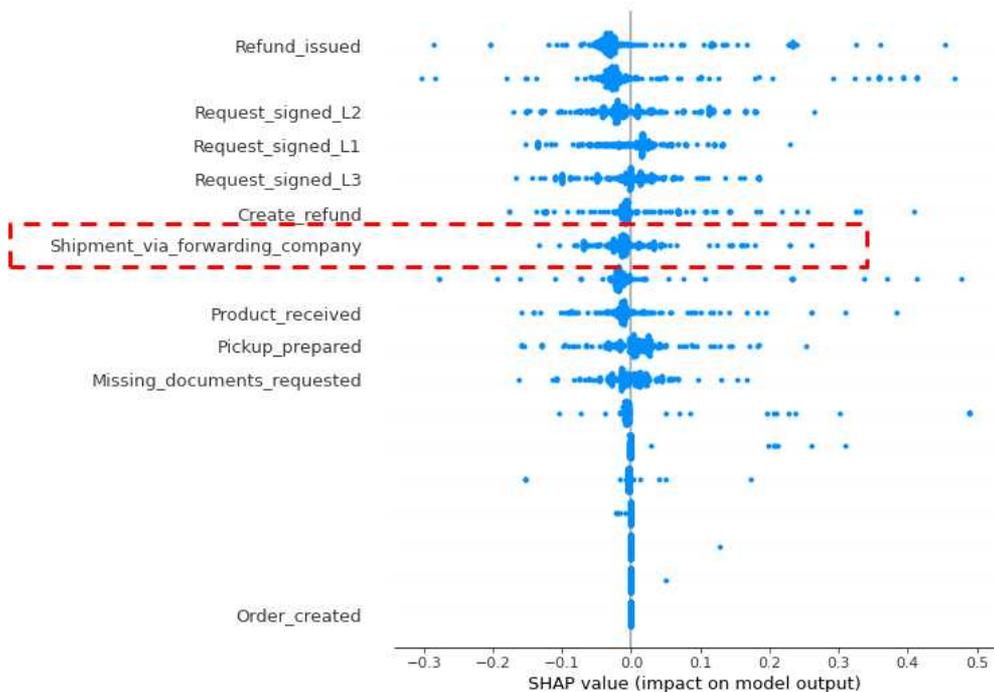
Rank	Activity	Contribution
1	Order Created	15.53715
2	Request Rejected L1	2.43365
3	Repairshop-X-Rejected	2.012444
4	Pickup Prepared	1.052724
5	<b>Shipment via Forwarding Company</b>	<b>1.036031</b>
6	Order Sent Back	0.904899
7	Request Signed L1	0.876655
8	Repair Shop-X-OK	0.812951
9	Special Refund Issued	0.762597
10	Product Received	0.754658
11	Amount Modified	0.067975
12	Warehouse	0.057908
13	Product Checked	-0.00115
14	Invoice Modified	-0.01171
15	Canceled	-0.0139
16	Supplements Requested	-0.02884
17	Shipment via Logistics Partner	-0.14689
18	Checked	-0.1489
19	Invoice Requested	-0.32605
20	Request Rejected L2	-0.68165
21	Repairshop-X-Checking	-1.36764

같은 예측 모델에 이번에는 SHAP 알고리즘을 적용해보았다. shap.DeepExplainer<sup>38)</sup>를 이용해 Explainer를 생성한 후 Test 데이터를 이용

해 Shapley Value를 구하였다. 생성된 Shapley Value를 이용해 임의의 Case에 대해 [그림 4-20]과 같이 Force Plot을 구한 결과 'Shipment via Forwarding Company'가 지연에 영향을 주는 기여도는 2번째로 높게 나타났다.



[그림 4-20] SHAP 알고리즘 적용 결과 (Force Plot)  
 [Fig. 4-20] Result Using SHAP Algorithm (Force Plot)



[그림 4-21] SHAP 알고리즘 적용 결과 (정찬일, & 이후진, 2020)  
 [Fig. 4-21] Result Using SHAP Algorithm (Summery Plot)

38) <https://shap-lrjball.readthedocs.io/en/latest/generated/shap.DeepExplainer.html>

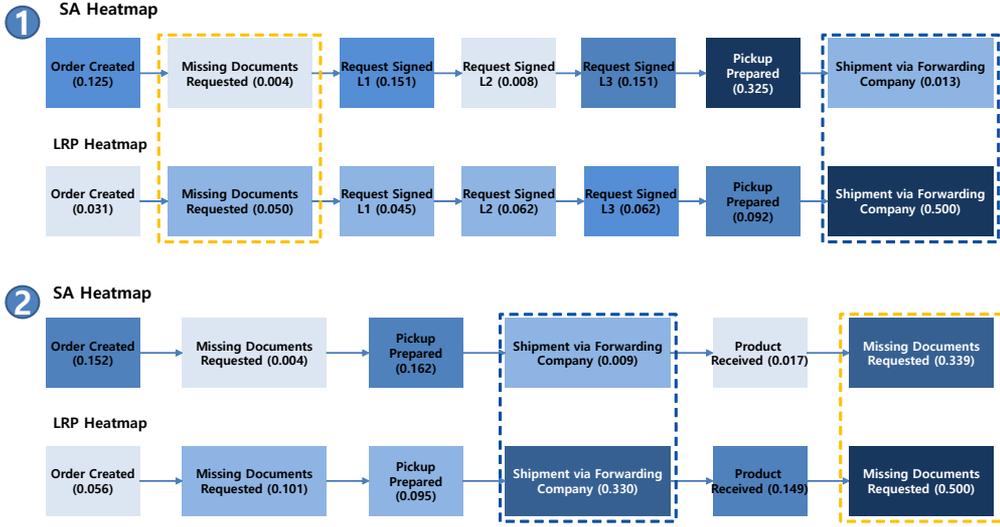
그러나 다시 Summary Plot을 이용해 전체적인 영향도를 확인한 결과는 [그림 4-21]과 같이 지연될 확률을 높인다는 방향성에서는 일치하였으나 기여도의 크기는 7번째로 오히려 LIME 알고리즘보다도 낮게 나타났다. 단순히 한두 번의 테스트 결과만으로 알고리즘 간의 우위를 논하기는 어렵지만, 일단 실험의 대상이 되는 프로세스 데이터에 대해서만큼은 SHAP 알고리즘이 LIME 알고리즘보다 우월하다고는 할 수 없는 결과이다.

마지막으로 같은 예측 모델에 SA와 LRP 알고리즘을 적용해보았다. LRP 알고리즘은 출력에서 입력 바로 전 단계까지의 모든 뉴런에 대해 활성화 함수  $a$ 와 가중치  $w$ 의 곱을 역전파 형태로 계산하는 방식을 통해 최초 입력값 별로 결과에 대한 기여도를 계산해낼 수 있다. 다만 이러한 방식은 은닉층이 존재하는 딥러닝 모델에서만 적용 가능한 알고리즘이다. [그림 4-17]은 SA와 LRP 알고리즘을 이용해 각각의 Activity 별 영향도가 높을수록 진한 색깔로 표시한 Heat Map이다. 원래 이미지 인식 모델로 개발된 LRP 알고리즘을 Montavon et al. (2017)에 기반하여 LSTM 등 Time Series에도 사용할 수 있도록 Open XAI Project에서 공개한 소스<sup>39)</sup>를 기반으로 재개발하여 프로세스 마이닝 예측 모델에 적용하였다.

[그림 4-18]의 Heat Map을 보면 SA 알고리즘의 예측 결과는 LIME 알고리즘의 예측 결과와 거의 유사한 결과를 보였으며, LRP 알고리즘은 [그림 4-18]의 ①의 Case에서는 ‘운송회사를 통한 배송’(‘Shipment via Forwarding Company’)의 기여도가 좀 더 높게 분석되어 정확도가 높았다. 그러나 [그림 4-18]의 ② Case에서는 프로세스 흐름의 뒤쪽에 배치된 Activity들에게 더욱 높은 가중치를 부여하는 편향이 발견되었다. 이러한 편향은 [그림 4-22]의 Case뿐만 아니라 100개 이상의 데이터를 테스트해보았을 때에도 거의 모든 Case에서 유사하게 나타났다.

---

39) <https://github.com/OpenXAIProject/tutorials/tree/master/LRP-Time-Series>



[그림 4-22] SA와 LRP 알고리즘 적용 결과 (정찬일, & 이후진, 2020)  
 [Fig. 4-22] Result Using SA & LRP Algorithm

$$R_d = \left( \frac{\partial f_c}{\partial x_d}(x) \right)^2 \quad (4-1)$$

$$R_j = \sum_k \frac{a_j w_{jk}^+}{\sum_j a_j w_{jk}^+} R_k \quad (4-2)$$

SA 알고리즘은 수식 (4-1)과 같이 입력값  $x_d$ 를 Target Class에 대한 예측함수  $f_c$ 에 대한 영향도로 직접 계산하는 Model-Agnostic 성격을 갖고 있으므로 LIME에서 도출된 결과와 매우 유사할 수밖에 없다. 그러나 LRP 알고리즘은 수식 (4-2)와 같이 출력에서 시작해서 전체 단계의 모든 뉴런에 대해 활성화 함수  $a_j$ 와 가중치  $w_j$ 의 곱을 계산하여 모델의 역전파 가중치를 계산하기 때문에 전반적으로 나중에 수행된 Activity에 더 많은 영향도를 주는 문제가 발생한 것으로 보인다.

결론적으로 딥러닝 알고리즘 (LSTM)에 기반한 프로세스 예측 모델에 대해 LIME, SHAP, SA, LRP 알고리즘을 이용한 원인분석 결과는 큰 편차가

없었으며, LIME 알고리즘이 상대적으로 우수한 성능을 보여주었다. 다만 이슈의 원인인 “Shipment via Forwarding Company”의 영향도가 3~5위로 나타나므로 사람이 개입하여 추가적인 분석을 해야만 정확한 결론을 내릴 수 있는 수준이었다. 물론 이런 결과가 나온 이유는 시간 지연을 예측하는 데에 Activity의 시간 데이터를 사용하지 않아 발생하는 예측 정확도가 낮은 문제와 해당 Case의 특수성이 있을 수 있다. 실제로 정찬일, & 이후진 (2019b)에서 테스트한 프로세스 결과는 의미 있는 정확도를 보여주었고, Jung, & Lee (2020)에서는 추가적인 Attribute를 통해 정확도를 높였다. 다만 이번 실험의 목표는 정확도를 높이는 것보다는 같은 기준 아래에 각각의 알고리즘 간의 성능을 측정하는 것이므로 각각의 Activity 간 걸린 시간 등을 추가하지 않고 진행하였다.

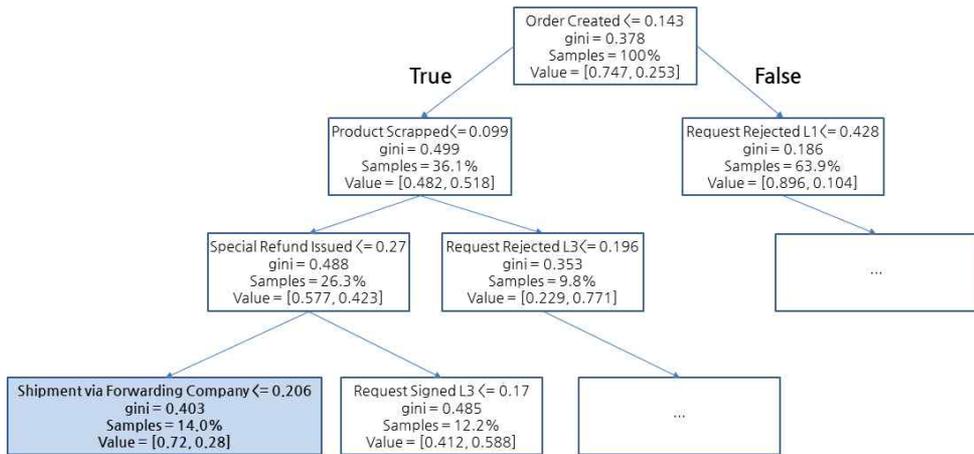
#### 4) 머신러닝 기반 예측 및 XAI 알고리즘 적용

이번에는 딥러닝 알고리즘 대신 보다 전통적인 머신러닝 알고리즘인 Decision Tree, Random Forest, Multinomial Naïve Bayes, Logistic Regression을 활용하여 예측 모델을 개발하였다. 4가지 모델에 동일한 데이터를 사용해 테스트해본 결과 [표 4-17]와 같이 Multinomial Naïve Bayes 예측 모델의 정확도 (Accuracy)가 0.8432로 가장 높았으나 전체적으로는 0.81 ~ 0.84 수준으로 큰차이가 없었고 이전에 실험했던 LSTM보다는 조금 낮은 정확도를 보였다.

[표 4-17] 예측 모델별 예측 정확도 비교 (정찬일, & 이후진, 2020)  
[Table 4-17] Accuracy Comparison by Machine Learning Models

Model	Accuracy
Decision Tree	0.8134
Random Forest	0.8283
Multinomial Naïve Bayes	0.8432
Logistic Regression	0.8320
LSTM	0.8654

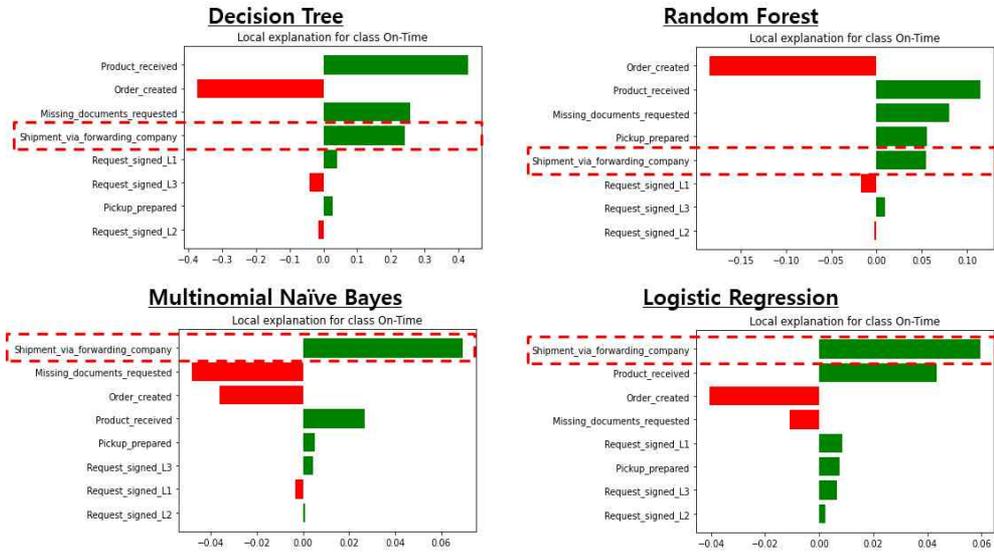
머신러닝에 대한 일반적인 오해 중 하나는 모든 머신러닝이 블랙박스 모델이라고 생각하는 것이다. 수많은 은닉층이 중첩되어 존재하는 딥러닝에서는 사실상 각각의 뉴런에서 가지고 있는 가중치 값들을 계산하는 것이 거의 불가능하므로 블랙박스라는 것이다. 그러나 Baehrens et al. (2010)에 따르면 Decision Tree 모델 학습으로 생성된 모델의 구조와 가중치 값 등을 직접 확인할 수 있는 화이트박스 모델이다. [그림 4-23]는 Decision Tree 모델의 내부 구조를 분석한 것으로 “Shipment via Forwarding Company” Activity는 4번째 노드에서 분류되며 지니 불순도 (gini)는 0.403이고, 14%의 샘플 데이터가 선택되었다는 정보 등을 추가로 알 수 있다.



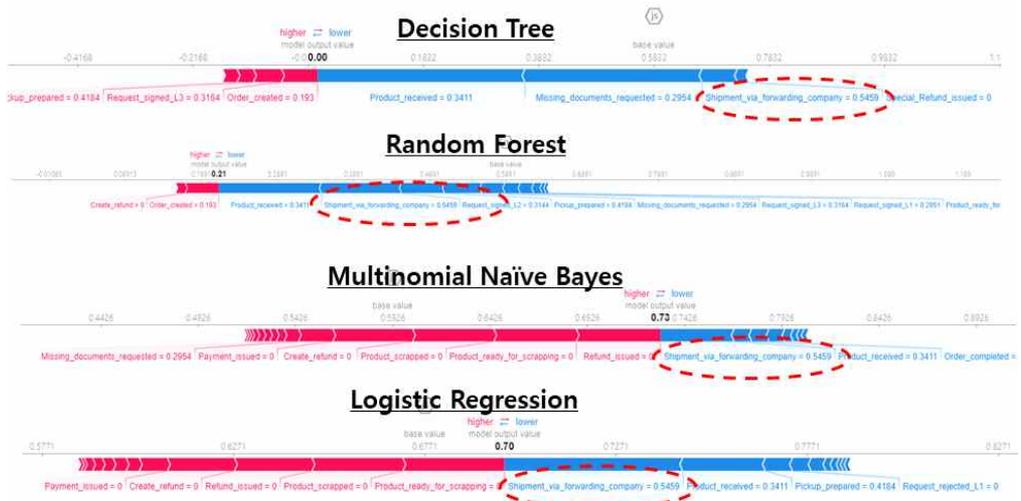
[그림 4-23] Decision Tree 모델의 가중치 (정찬일, & 이후진, 2020)  
[Fig. 4-23] Decision Tree Model Weight Composition

각각의 머신러닝 모델에서 ‘지연’ (‘Delay’)으로 판단된 임의의 Case를 LIME 알고리즘을 이용하여 분석해본 결과 [그림 4-24]과 같이 ‘Shipment via Forwarding Company’가 지연에 대한 원인이라는 방향성은 모든 모델에서 일치하였으나 Decision Tree와 Random Forest 모델은 영향도가 3위와 4위로 LSTM과 유사한 수준이며 Multinomial Naïve Bayes 모델과 Logistic Regression 모델은 모두 영향도가 1위로 ‘Shipment via forwarding company’ Activity가 ‘지연’ 발생의 가장 큰 원인이라고 분석하여 유의미한

결과를 보여주었다.



[그림 4-24] LIME 기반 해석 결과 (정찬일, & 이후진, 2020)  
 [Fig. 4-24] LIME based Analysis Results for Learning Models



[그림 4-25] SHAP 기반 해석 결과 (정찬일, & 이후진, 2020)  
 [Fig. 4-25] SHAP based Analysis Results for Learning Models

마찬가지로 같은 샘플 데이터에 SHAP 알고리즘을 적용해 본 결과도 [그림 4-25]처럼 ‘Shipment via Forwarding Company’의 지연에 대한 영향도가 Decision Tree는 3위, Random Forest 모델은 2위, Multinomial Naïve Bayes 모델과 Logistic Regression 모델은 모두 1위로 나타나서 LIME 알고리즘과 거의 유사한 것을 확인할 수 있었다.

결론적으로 동일한 조건이라면 딥러닝 모델보다 머신러닝 기반의 예측 모델에 설명 가능한 인공지능을 적용하는 것이 이슈 원인분석을 하는데, 더욱 유리하며 특히 Multinomial Naïve Bayes 모델과 Logistic Regression 모델이 상대적으로 안정적인 결과를 보여준다는 것을 확인하였다.

#### 5) 프로세스 흐름 기반 영향도 분석

일반적으로 프로세스 모델의 특징은 개별 Activity 자체보다는 Activity 간의 흐름이 중요하다는 데 있다. 이러한 특징에 기반하여 위의 실험에서 가장 높은 해석력을 보여준 Logistic Regression 기반의 예측 모델에 대해 개별 Activity 대신 연결된 Activity 흐름 단위로 영향도를 측정하였다. 즉 개별 Activity가 아닌 2개 이상의 Activity 간 흐름을 인자로 하여 영향도를 분석하면 동일한 알고리즘임에도 불구하고 [표 4-18]과 같이 “Shipment via Forwarding Company”가 포함된 “Shipment via Forwarding Company → Product Received”의 기여도가 가장 높았으며 “Pickup Prepared → Shipment via Forwarding Company”까지 포함하면 다른 Activity에 비해 월등히 강한 영향을 준다는 것을 확인할 수 있었다. 또한, 부가적으로 “Missing Documents Requested → Missing Documents Requested”처럼 같은 업무가 반복될 때 부정적인 영향을 준다는 것까지 확인할 수 있어 업무 반복 (Loop)에 대한 성능 이슈까지 한 번에 분석할 수 있다.

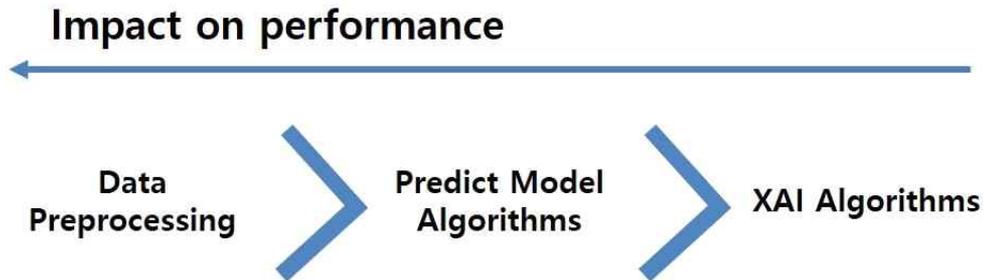
[표 4-18] Activity 흐름 기반 영향도 분석 (정찬일, & 이후진, 2020)  
 [Table 4-18] Contribution Including Activity Flow

Contribution	Feature
0.897	Shipment via Forwarding Company → Product Received
0.746	Missing Documents Requested → Missing Documents Requested
0.739	Create Refund → Refund Issued
0.642	Pickup Prepared → Shipment via Forwarding Company
0.623	Order Created → Missing Documents Requested
0.501	Product Received → Create Refund
0.435	Request Signed L3 → Pickup Prepared
0.393	Request Signed L2 → Request Signed L3
0.386	Missing Documents Requested → Request Signed L1
0.373	Request Signed L1 → Request Signed L2

## 6) 실험 결과 및 의미

본 실험에서는 프로세스 결과 예측을 위해 딥러닝 모델인 LSTM과 Decision Tree, Random Forest, Logistic Regression, Multinomial Naïve Bayes 등 머신러닝 모델을 적용해보았다. 또한, 각각의 예측 모델에 대해 LIME, SHAP, SA, LRP 등의 다양한 XAI 알고리즘을 적용한 결과를 실험하였다. 실험 결과를 보면 예측 모델 자체의 정확도는 딥러닝 기반의 LSTM 모델이 가장 높았으나 원인분석에 대한 해석력 측면에서는 전통적인 머신러닝 모델인 Logistic Regression이 가장 높게 나왔고 Decision Tree는 모델 내부의 구조를 확인할 수 있는 화이트 박스 분석이 가능하기 때문에 예측 모델의

정확도와 예측 결과에 대한 해석력이 일정 부분 Trade-Off 관계를 맺으며 최종 분석 결과에서 영향을 준다는 것을 알 수 있었다. 그러나 설명 가능한 인공지능 알고리즘 간에는 원인분석 결과에 대해 유사한 결과를 보여주고 있어 유의미한 차이가 없었다. 즉 설명 가능한 인공지능 알고리즘보다는 분석 대상이 되는 머신러닝 모델의 특성이 오히려 최종적인 해석력에 더 많은 영향을 준다는 것을 확인할 수 있었다. 또한, 프로세스 모델의 특성상 각각의 개별 Activity가 아닌 2개 이상의 Activity 간 흐름을 대상으로 실험하였을 때 더욱 높은 해석력을 얻을 수 있었다. 다시 말해 [그림 4-26]와 같이 이슈 원인분석 자동화의 성능에는 설명 가능한 인공지능 알고리즘보다는 예측 모델이, 예측 모델의 알고리즘보다는 프로세스 데이터에 대한 전처리가 더 높은 영향을 미친다고 할 수 있다.



[그림 4-26] 이슈 원인분석 성능에 대한 영향도  
 [Fig. 4-26] Impact on Causal Analysis Performance

### 제 3 절 모바일 앱 사용 프로세스 분석

#### 1) 모바일 앱 프로세스 특성 및 전문가 분석 결과

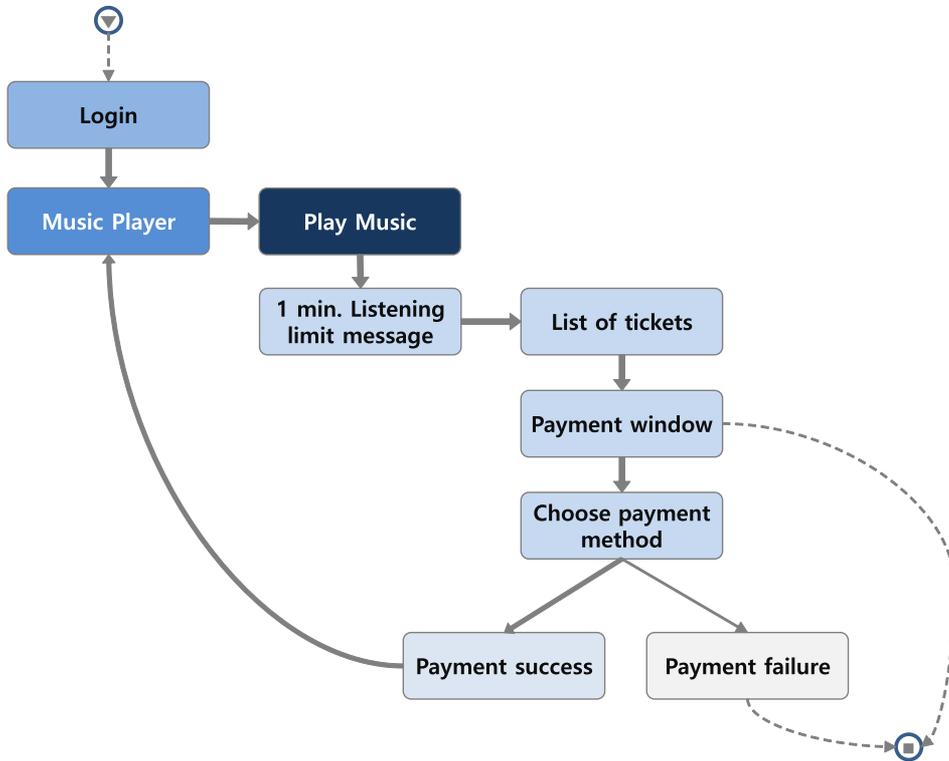
Loen Entertainment, & PMIG (2017)에 따르면 M사는 스마트폰 보급과 함께 급성장한 대한민국의 디지털 음원 서비스 시장의 50% 이상을 점유하여,

2,800만명 이상의 가입자와 400만명 이상의 유료 음원 이용자를 확보한 대한민국의 최대의 음원 서비스 기업이다. K사와의 합병 후 M사는 가입자 확대를 위해 K메신저 이용자는 별도 가입 없이 음원을 들을 수 있도록 하고 이 사용자들을 유료 가입자로 전환하기 위한 캠페인을 실행하였다. 이를 위해 K메신저 계정으로 유입된 사용자의 행동을 이해하고, 이들의 구매전환율 (CVR: Conversion Rate)을 높이기 위한 프로세스 마이닝 프로젝트를 수행했다. 프로젝트팀은 모바일 앱 사용자의 서비스 사용을 기록한 시스템 로그 데이터를 추출한 후 추출된 데이터에 한글로 작성한 메뉴 이름을 붙여주는 전처리를 하였다. 전처리된 파일을 프로세스 마이닝 도구 (DISCO)에 로딩한 다음에 “User Sequence Number”와 “Menu Name”을 각각 Case ID와 Activity로 설정하여 프로세스 맵을 생성했으며, 기업기밀을 제외한 일부 데이터를 학습 및 연구용으로 공개하였다.

프로세스 마이닝의 주요 목적인 K메신저를 통해 가입한 사용자의 가입 당일에 모바일 앱을 사용한 이동 동선 (Customer Journey)을 대상으로 프로세스 분석을 진행하여 ①서비스 가입자 증대, ②신규 가입자의 구매전환율 (CVR) 제고, ③행동 패턴에 따른 신규 가입자 분류 및 가입 의도 파악을 수행하였다. 프로젝트팀은 프로세스 마이닝을 통해 프로세스 모델을 자동으로 생성하였고 이를 분석하여 신규 가입자를 5가지 고객군으로 분류하였고, 각 고객군의 가입목적과 핵심 화면을 파악하였다. 또한, 고객 유형별 회원가입 목적에 맞춤형 타겟 마케팅 캠페인을 각 유형의 고객이 자주 방문하는 페이지에서 시행하도록 제시하였으며, 목표로 한 구매전환율 (CVR)이 달성되지 않은 원인을 파악하여 대안을 제시했다 (Loen Entertainment, & PMIG, 2017).

특히 구매전환율이 낮은 고객군의 특징 중 하나인 지불 오류에 대한 원인을 분석하기 위해 [그림 4-27]과 같이 프로세스 마이닝 도구를 사용하여 프로세스 모델을 생성하고 전체 지불 시도 사례에서 성공한 프로세스와 실패한 프로세스의 차이를 비교했다. 프로젝트팀의 프로세스 마이닝 전문가들은 다양한 통계 분석 및 업무 경험에 따른 직관을 통해 티켓 결제 실패의 원인은 특정 OS 및 앱 버전 문제라는 사실을 알아냈으며 그 결과를 앱 개발 관리자에

게 통보하여 해당 앱 버전의 Patch를 진행하여 문제를 해결했다.



[그림 4-27] 음원 서비스 결제 프로세스 (Jung, & Lee, 2020)  
 [Fig. 4-27] Music Streaming App's Payment Process Discovery

## 2) 데이터 전처리

프로세스 마이닝 전문기업인 PMIG는 모바일 앱 사용자의 서비스 이용을 기록한 로그 데이터를 허둡 시스템(Hadoop System)에서 추출하여 프로세스 마이닝을 진행하였으며, 데이터 일부를 가공하여 교육목적으로 공개하였다. 해당 데이터 파일은 [표 4-19]와 같이 52,657개의 Case와 993,751개의 이벤트 (Activity)로 구성되어 있으며, 프로세스 구성에 필수적인 Case ID, Activity, End Time을 포함해 나이, App 버전, 성별, OS 버전 등의 추가적인 분석을 위한 속성 (Attribute)들도 포함되어 있다. 모바일 앱의 사용자 로그를

기반한 고객 동선을 Customer Journey라고 한다. 이러한 Customer Journey 데이터는 기존의 업무 프로세스와 달리 평균 4.3분에 불과할 정도로 단기간에 진행된다는 특징을 갖고 있다. 또한, 기기별 식별정보를 이용해 OS Version 및 App Version을 자동으로 파악할 수 있다. 다만 Age, Sex 등의 사용자 정보는 로그인한 사용자에 대한 것만 파악할 수 있으므로 많은 Event에서 비어 있기도 하다. 또한, 일반 업무 프로세스와 비교하면 표준이 없으므로 동일하게 수행되는 경우가 매우 적다. 그래서 같은 순서대로 수행된 Case들의 집합인 Variant의 숫자와 Case의 숫자가 별로 차이가 나지 않는다.

[표 4-19] 모바일 앱 사용자 로그 데이터 세트 특성  
[Table 4-19] Mobile App Process Data Set Feature

Data Feature	Count
Number of Variants	45,004
Number of Cases	52,657
Number of Events	993,750
Distinct Activities	42
Median Case Duration	35 Secs
Mean Case Duration	4.3 Mins
Age	51 (12-62)
App Version	24 (4.0.4 - 4.7.1)
Sex	2 (M/W)
OS Version	46 (9.0 - 12.0)

전체 42개 Activity들의 수행 빈도와 특징은 [표 4-20]과 같이 각각의 App 기능 명칭과 같으며 가장 많이 사용되는 기능인 음악재생의 발생 빈도(Frequency)가 가장 높다. 다만 특이한 것은 로그인 Event가 수행된 비율이 5.03%에 불과하여 로그인하지 않고 미리 듣기만 이용하는 사용자의 숫자가 꽤 많음을 알 수 있다.

[표 4-20] 모바일 앱 Activity 특성  
 [Table 4-20] Mobile App Activity's Feature

Activity	Frequency	Relative frequency
음악재생	229,566	23.10%
이용권 확인	114,453	11.52%
음악재생요청	114,428	11.51%
음악 플레이어	76,733	7.72%
실시간차트	66,103	6.65%
로그인	50,000	5.03%
검색	21,581	2.17%
고객지원	18,898	1.90%
자주하는질문	18,289	1.84%
이용권목록보기	17,935	1.80%
유형별로찾기_이벤트	17,422	1.75%
인디음악	17,243	1.74%
유형별로찾기_서비스문의/오류	17,232	1.73%
1:1문의하기	17,186	1.73%
유형별로찾기_이용권사용문의	17,186	1.73%
유형별로찾기_기타	17,125	1.72%
장르음악	17,064	1.72%
발라드	16,950	1.71%
랩/힙합	16,901	1.70%
유형별로찾기_아지툽	16,875	1.70%
유형별로찾기_회원정보	16,863	1.70%
트로트	16,782	1.69%

R&Soul	16,768	1.69%
Rock	16,757	1.69%
포크/블루스	16,725	1.68%
최초로그인	2,657	0.27%
1분미리듣기 제한메시지	2,460	0.25%
콜센터연결	1,039	0.10%
이벤트참여	1,013	0.10%
유형별로찾기_결제/해지/환불	596	0.06%
앨범댓글작성	407	0.04%
이달의가수투표	379	0.04%
음악방송방청권신청	379	0.04%
어학서비스이용	377	0.04%
콘서트티켓예매	371	0.04%
노래방서비스이용	364	0.04%
동영상재생	195	0.02%
결제창진입	149	0.01%
결제수단선택	149	0.01%
이용권구입완료	133	0.01%
결제실패	16	0%
동영상목록보기	2	0%

또한, 전체 데이터에서 App Version이 포함된 데이터를 분류해보면 [표 4-21]처럼 사용자 대부분이 4.6.0 이상을 사용하고 있으나, 소수의 사용자는 아직도 4.6.0 이하의 버전을 사용하고 있는 것을 알 수 있다. 데이터 중 ‘.4.3.6’은 4.3.6이 잘못 입력된 것으로 판단되어 예측 모델에서는 ‘4.3.6’으로 변경해 사용하였다.

[표 4-21] 모바일 앱 버전 데이터 특성  
 [Table 4-21] Mobile App Version Data Feature

Value	Frequency	Relative frequency
.4.3.6	534	0.05%
4.0.4	554	0.06%
4.1.4	388	0.04%
4.1.8	451	0.05%
4.2.2	471	0.05%
4.2.6	440	0.04%
4.3.2	536	0.05%
4.3.8	590	0.06%
4.4.3	548	0.06%
4.4.4	630	0.06%
4.4.6	682	0.07%
4.4.8	438	0.04%
4.6.0	83,001	8.35%
4.6.1	81,917	8.24%
4.6.2	82,690	8.32%
4.6.3	81,915	8.24%
4.6.4	81,879	8.24%
4.6.5	84,244	8.48%
4.6.6	82,014	8.25%
4.6.7	81,050	8.16%
4.6.8	81,157	8.17%
4.6.9	82,801	8.33%
4.7.0	81,441	8.20%
4.7.1	83,380	8.39%

전체 데이터를 프로세스 마이닝 도구인 DISCO로 프로세스 모델을 생성한 후 Filter 기능을 통해 '티켓 결제'를 시도한 Case만 선택하여 Activity 중 '결제 성공'이라는 이벤트가 있을 경우는 'Success'로, '결제 실패' 이벤트가 발생한 경우는 'Fail'로 Labeling한 후 CSV 파일로 변환하였다. 이 파일에서 다시 Label이 'Success'인 경우는 1로, 'Fail'일 경우는 0으로 변환하고 결제 결과에 직접적인 영향을 미치는 Activity ('결제 성공', '결제 실패')를 삭제하였다. 로그 데이터에 등록된 Activity, App 버전, OS 버전, 나이, 성별, 결제 수단 등의 속성은 전체 출현하는 경우의 수를 벡터의 차원으로 하고, 해당하는 속성의 칼럼값에 1의 값을 부여하고, 다른 칼럼값에는 0을 부여하는 방식으로 원-핫 인코딩 (One-Hot Encoding) 처리하여 벡터 데이터로 변환하였다. 정찬일, & 이후진 (2020)에서 확인한 바와 같이 원인분석 자동화의 성능을 향상하기 위해 기존의 실험에서는 Activity의 순서를 기반으로 Text Mining 기법과 유사하게 Word Embedding 형태로 사용한 것과 달리 원-핫 인코딩으로 처리하고 Activity 외에도 다양한 추가 속성 (Attribute)을 추가함으로써 결과 예측 및 이슈 원인에 대한 해석력이 향상되도록 하였다. 원-핫 인코딩 처리한 후의 데이터 Column은 다음과 같이 72개로 분할되었으며 모든 전처리 작업이 끝난 후의 최종 데이터는 [표 4-22]와 같이 구성되었다.

[표 4-22] 모바일 앱 로그 데이터 전처리 결과  
 [Table 4-22] Mobile App Log Data Preprocessing

No	Activity _1분미리 듣기제한 메시지	Activit y_검색'	Activity _결제수 단선택	...	나이	지불수 단_간편 결제	지불수단 _신용카 드결제
1	1	0	0	...	25	0	0
2	0	1	0	...	25	0	0
3	0	0	1	...	25	0	1
...							

['Activity\_1분미리듣기 제한메시지', 'Activity\_검색', 'Activity\_결제수단선택', 'Activity\_결제실패', 'Activity\_결제창진입', 'Activity\_동영상목록보기', 'Activity\_동영상재생', 'Activity\_뮤직플레이어', 'Activity\_실시간차트', 'Activity\_앨범댓글작성', 'Activity\_음악재생', 'Activity\_이벤트참여', 'Activity\_이용권구입완료', 'Activity\_이용권목록보기', 'Activity\_최초로그인', 'App버전\_4.1.4', 'App버전\_4.1.8', 'App버전\_4.2.2', 'App버전\_4.2.6', 'App버전\_4.3.8', 'App버전\_4.4.3', 'App버전\_4.4.4', 'App버전\_4.4.6', 'App버전\_4.6.0', 'App버전\_4.6.1', 'App버전\_4.6.2', 'App버전\_4.6.3', 'App버전\_4.6.4', 'App버전\_4.6.5', 'App버전\_4.6.6', 'App버전\_4.6.7', 'App버전\_4.6.8', 'App버전\_4.6.9', 'App버전\_4.7.0', 'App버전\_4.7.1', 'Case ID', 'OS버전\_10.0.2', 'OS버전\_10.0.3', 'OS버전\_10.1', 'OS버전\_10.3.1', 'OS버전\_10.3.3', 'OS버전\_11.0', 'OS버전\_11.0.1', 'OS버전\_11.0.1(15A8391)', 'OS버전\_11.0.2', 'OS버전\_11.0.3', 'OS버전\_11.1', 'OS버전\_11.1.1', 'OS버전\_11.1.2', 'OS버전\_11.2', 'OS버전\_11.2.1', 'OS버전\_11.2.2', 'OS버전\_11.2.5', 'OS버전\_11.2.6', 'OS버전\_11.3', 'OS버전\_11.3.1', 'OS버전\_11.4', 'OS버전\_11.4.1', 'OS버전\_12.0', 'OS버전\_9.0.2', 'OS버전\_9.1', 'OS버전\_9.2.1', 'OS버전\_9.3(13E236)', 'OS버전\_9.3(13E237)', 'OS버전\_9.3.1', 'OS버전\_9.3.2(13F72)', 'OS버전\_9.3.3', 'OS버전\_9.3.4', 'OS버전\_9.3.5', '나이', '지불수단\_간편결제', '지불수단\_신용카드결제']

마지막으로 각각의 Activity를 프로세스별로 시간의 흐름에 따라 정렬하여 52,657개의 Case 별로 43개의 시간 단계와 71개의 인코딩 된 특징 (Feature) 으로 구성된 3차원의 시계열 (Time-Series) 데이터를 생성했다.

### 3) 예측 모델 작성

예측 모델은 [그림 4-28]과 같이 Tensorflow를 사용하여 1,024개의 LSTM Unit과 2개의 Dense Unit으로 구성하였으며, 4,495,362개의 파라미터

를 가진 LSTM 알고리즘으로 설계하였다.

---

Layer (type)	Output Shape	Param #
lstm_first (LSTM)	(None, 1024)	4493312
dropout_first (Dropout)	(None, 1024)	0
dense_first (Dense)	(None, 2)	2050

---

Total params: 4,495,362  
Trainable params: 4,495,362  
Non-trainable params: 0

---

[그림 4-28] 예측 모델 구성 파라미터 (Jung, & Lee, 2020)  
[Fig. 4-28] Predict Model's Parameters

전처리 된 데이터를 통해 학습한 결과, 티켓 구매 과정의 '성공'과 '실패'의 예측 정확도는 96% (0.9667)로 비교적 우수하게 결과를 예측하는 것을 확인하였다.

#### 4) 이슈 원인분석 결과

대표적인 XAI 알고리즘인 LIME를 사용하여 예측 모델을 분석한 결과 예측 결과에 대한 각 특성의 기여도를 수치로 식별할 수 있었다. 다양한 프로세스가 진행되어 모든 Case를 한 번에 확인하기 어려우므로 먼저 가장 빈번하게 발생하는 7단계 프로세스인 '로그인 (Login)' → '뮤직플레이어 (Music Player)' → '음악재생 (Play Music)' → '음악듣기 제한 메시지 (Listening Limit Message)' → '이용권 확인 (List of Ticket)' → '결제창 (Payment Window)' → '지불방법선택 (Choose Payment Method)'를 정하고 여러 프로세스에서 수집된 각 단계별 Activity 및 Attribute (App버전, OS 버전, 나이, 성별, 지불수단 등)의 기여도를 계산하였다.

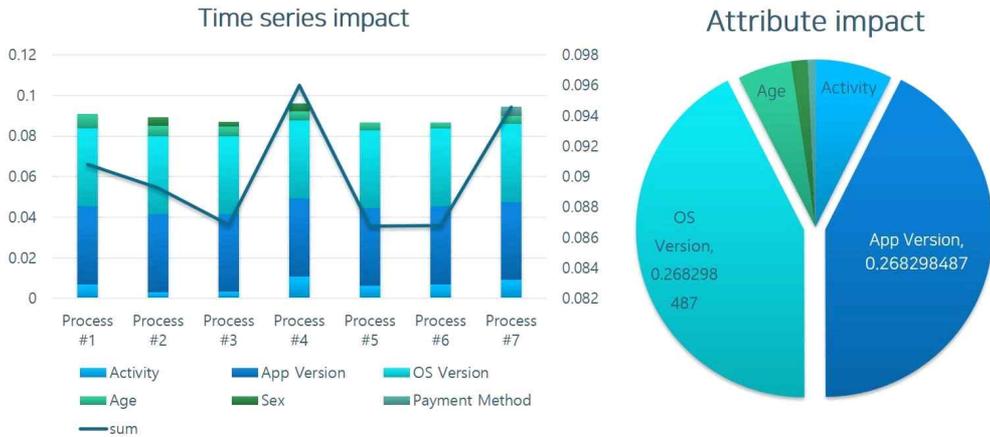
[표 4-23] 결제 실패 프로세스의 원인분석 결과 (Jung, & Lee, 2020)  
 [Table 4-23] Failed Process Causal Analysis Automation Result

Attributes	Step #1	Step #2	Step #3	Step #4	Step #5	Step #6	Step #7
	0.09081602	0.089235	0.086856	0.095998	0.086729	0.086789	0.094582
Activity 0.046949	Login 0.00708	Music player 0.00313	Play music 0.00329	listening limit message 0.01093	List of ticket 0.00619	Payment window 0.00708	Choose payment method 0.00926
App Version 0.268298	4.4.3 0.03833	4.4.3 0.03833	4.4.3 0.03833	4.4.3 0.03833	4.4.3 0.03833	4.4.3 0.03833	4.4.3 0.03833
OS Version 0.268298	10.0.2 0.03833	10.0.2 0.03833	10.0.2 0.03833	10.0.2 0.03833	10.0.2 0.03833	10.0.2 0.03833	10.0.2 0.03833
Age 0.032666	35 0.00708	35 0.00542	35 0.0047	35 0.00447	35 0.00389	35 0.00305	35 0.00407
Gender 0.01019	W 0.00000	W 0.00403	W 0.00222	W 0.00394	W 0.00000	W 0.00000	W 0.00000
Payment Method 0.004603							Credit card 0.00460



[표 4-23]에서 분석된 프로세스는 '결제 실패' 확률이 0.91 (실제로 결과도 '실패')이며, 예측 결과에 대한 각 Feature (Activity와 Attribute)의 기여도, 즉 결제 실패에 대한 원인이 될 수 있는 확률을 계산한 결과이다. 가로축인 Step은 Activity의 시계열 속성으로 수행된 순서를 나타내며 Activity 외에

App Version, OS Version, Age, Gender, Payment Method는 모두 Attribute 들로 원인분석의 효율을 높이기 위해 포함되었다. 단 모든 Activity는 동일한 사용자의 데이터이므로 대부분의 Attribute는 같은 값을 갖지만 ‘결제 수단’ (Payment Method)의 경우는 마지막 ‘Choose Payment Method’ Activity에 서만 나타나는 속성으로 그 이전의 Activity에서는 모두 공란으로 처리되었다.

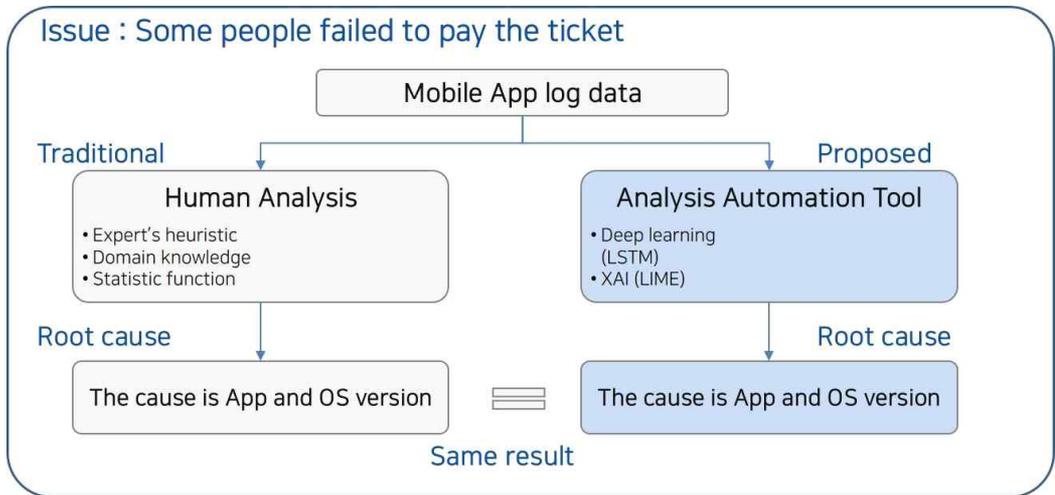


[그림 4-29] 요소별 영향도 (Jung, & Lee, 2020)  
 [Fig. 4-29] Activity & Attribute's Impact

Step #4의 경우 Listening Limit Message가 예측 결과 (실패)에 0.01093의 영향을 주어 다른 Step에 비해 상대적으로 큰 0.095998의 영향도를 갖고 있으나 다른 Step들과의 차이가 0.01 정도로 크지 않았다. 그러나 세로축인 속성항목 (Attribute)의 경우는 App 버전과 OS 버전의 예측 결과에 대한 영향도가 모든 단계에 걸쳐 높은 결과를 보여주고 있으며 합계가 각각 0.268298로 나이 (0.032666)나 성별 (0.01019)에 0.17 이상 높다. 즉 [그림 4-29]에서 집계된 바와 같이 전체 프로세스 수행 결과 예측에 가장 많은 영향을 미치는 특징은 App 버전과 OS 버전이라는 것을 명확하게 알 수 있다. 다시 말하면 App 버전과 OS 버전이 결제 실패의 주요 원인이라는 결과가 사람의 개입 없이 자동으로 도출되었다.

## 5) 실험 결과 및 의미

알고리즘을 통해 자동으로 도출된 결과는 '결제 실패'의 근본 원인이 특정 앱 버전 (4.4.3)과 OS 버전 (10.0.2)임을 나타내고 있다. 이 결과는 Loen Entertainment, & PMIG (2017)이 프로세스 마이닝 전문가들의 분석으로 찾아낸 '결제 실패'의 원인과 같다.



[그림 4-30] 전문가와 알고리즘 분석 결과 비교(Jung, & Lee, 2020)  
[Fig. 4-30] Human and Proposed Algorithm's Analysis Are Same

[그림 4-30]에서 볼 수 있듯이 '일부 사용자들의 이용권 결제에 실패'하는 이슈에 대해 기존의 프로세스 마이닝은 모바일 앱의 프로세스 모델을 전문가의 직관을 통해 분석하여 App과 OS의 버전이 원인이라는 결론을 내렸다. 그러나 같은 모바일 앱 데이터를 인공지능을 통해 예측 모델을 만들고, 예측 결과가 '결제 실패'로 나오는데 기여한 요소 (Factor)를 설명 가능한 인공지능을 이용해 자동으로 도출한 결과 역시 App과 OS 버전의 문제라는 동일한 결론에 도달할 수 있었다. 이를 통해 제시한 설명 가능한 인공지능 기반의 이슈 원인분석 알고리즘이 실제 업무에서도 충분히 활용 가능하다는 것을 증명할 수 있었다.

## 제 5 장 결론

### 제 1 절 연구 요약

본 논문에서는 네덜란드의 R은행, 다국적 기업인 P전자회사, 그리고 한국의 음원 서비스 기업인 M사의 업무 프로세스 데이터를 기반으로 각각의 프로세스에 대한 수행 결과를 사전에 예측할 수 있는 머신러닝 기반의 예측 모델을 개발하였으며 이를 다시 설명 가능한 인공지능을 통해 사람의 개입 없이 이슈의 원인을 자동으로 도출하는 데 성공하였다.

은행의 여신 프로세스를 기반으로 고객이 대출 신청을 취소할지를 사전에 예측하는 딥러닝 모델을 개발하였다. 데이터 불균형을 해결하기 위해 학습데이터를 리샘플링한 후 LSTM 알고리즘을 사용해 학습한 결과 테스트 데이터의 정확도 (Accuracy)가 96%에 도달하였다. 이를 통해 신규 대출 신청이 접수되었을 때 프로세스 진행 경과를 모니터링하면 고객 취소를 사전에 예측하고 금리 우대 등의 적절한 프로모션을 통해 고객 취소율을 높일 수 있을 것이다. 또한, 이 예측 모델을 설명 가능한 인공지능 기반으로 원인을 분석한 결과 금리 조건을 고객에게 제시 후 고객과의 상담 전화 통화가 여러 차례 발생하거나 서류보완 작업이 발생할 때 고객이 신청을 취소할 확률이 높아지는 것을 확인하였다. 이 결과를 기반으로 은행에서는 고객과의 전화상담이 10회 이상 반복되거나 서류보완 작업이 반복될 때는 금리 우대 등을 통해 취소를 방지하거나, 반대로 불필요한 업무 노력을 방지하기 위해 해당 건을 자동 승인거절 처리하는 등의 업무 정책 (Rule)을 개선할 수 있을 것이다.

글로벌 전자제품 제조사의 환불 프로세스를 검토하여 환불 처리가 지연되는 원인을 찾기 위해 다양한 머신러닝 기반의 예측 모델과 설명 가능한 인공지능 알고리즘들을 조합하여 테스트해보았다. 실험 결과 전통적인 머신러닝 모델들에 비해 딥러닝 모델의 예측 정확도는 높았으나, 예측 결과에 대한 원인을 분석하는 데는 오히려 불리하다는 것을 확인하였다. 그러나 최종적으로 원인분석의 정확도에 결정적인 역할을 하는 것은 데이터의 전처리 방법에 달

려 있고 프로세스를 구성하고 있는 Activity 자체보다는 Activity 간의 흐름 (2개 이상의 Activity 조합)을 기준으로 원인을 분석하는 것이 가장 효율적이라는 결론을 얻을 수 있었다.

마지막으로 국내 모바일 음원 서비스 앱의 고객 여정 (Customer Journey) 프로세스를 분석하였다. 고객 여정 프로세스는 일반 업무 프로세스에 비해 짧은 시간에 수행되고 프로세스 흐름의 일관성도 매우 낮은 특징을 갖고 있다. 원인분석의 대상은 고객들이 이용권 결제를 진행하다 실패하여 이탈하는 경우의 원인을 찾는 것이었으며, 이를 위해 기존의 실험과 달리 Activity뿐만 아니라 사용자의 성별, 나이, OS 버전, App 버전, 결제 방법 등 다양한 속성들을 추가하여 분석하였다. 분석 결과 고객이 결제를 실패하는 가장 큰 원인은 특정한 OS 버전과 App 버전인 것으로 밝혀졌으며 이는 실제 프로세스 마이닝 프로젝트에서 전문가들이 참여해 밝혀낸 결과와 정확히 일치하였다.

실험을 통해 밝히고자 한 ‘알고리즘에 의해 자동으로 도출된 이슈 원인이 전문가들이 프로젝트를 통해 찾아낸 이슈 원인과 동일하다’는 가설은 위의 3가지 실험 모두에서 확인할 수 있었다. 또한, 각각의 알고리즘별 장단점과 데이터에 따른 특징들을 상세히 비교하였으며 정확한 분석을 위한 데이터 전처리 방법까지 제시하였다. 제안된 머신러닝 모델과 설명 가능한 인공지능 알고리즘을 활용하면 전문가의 개입 없이도 업무 이슈에 대한 원인을 자동으로 찾아낼 수 있으며, 단순 Activity만이 아닌 추가적인 속성들을 함께 학습할 때 더욱 정확한 원인분석이 가능하다는 결론을 내릴 수 있었다.

## 제 2 절 연구의 기여 및 시사점

본 연구를 통해 현행 업무 프로세스 모델을 자동 생성하는 용도 위주로 활용되던 기존의 프로세스 마이닝을 한 단계 발전시켜 사전에 프로세스의 수행 결과를 예측함으로써 업무 효율성을 높여줄 뿐만 아니라 프로세스 자체를 개선하기 위한 원인분석을 정량적으로 자동화할 수 있음이 증명되었다. 프로세스 이슈 원인분석을 자동화하였다는 것은 첫 번째로 기존의 업무혁신 프로

젝트들이 큰 비용과 인력이 소요되어 자주 진행되지 못하던 문제를 해결함으로써 상시적인 업무혁신 체계가 가능하며 이를 통해 기업의 경쟁력을 향상할 수 있음을 의미한다. 두 번째로는 전문가의 경험과 역량에 의존하지 않게 되어 프로젝트 참여 인력의 역량이 조금 떨어져도 언제나 같은 결과를 보장한다는 의미이며, 심지어 내부의 업무 담당자만으로도 업무혁신을 진행하는 것이 가능하다는 뜻이다. 마지막으로 이슈 원인을 사람의 감이 아닌 수치에 의한 정량적인 분석을 통해 알아낼 수 있으므로 보다 정확한 원인분석과 그 결과에 대한 명확한 근거 제시가 가능하게 된다. 결론적으로 프로세스 이슈 원인분석 자동화를 실제 기업에 적용함으로써 더 적은 비용과 인력으로, 더 정확하고 빠른 분석 결과를 통해, 상시적인 업무혁신을 수행할 수 있으며 이를 통한 기업의 성과 달성과 경쟁력 강화에 이바지하게 될 것이다.

특히 PI를 수행하기 위한 인력과 비용이 부족한 중소기업의 경우 정기적으로 컨설팅을 받기 어려운 환경으로 프로세스 마이닝과 본 연구에서 제시한 원인분석 자동화를 통해 자동화 관리 되는 것이 더욱 필요하다. 또한, 대기업과 같이 ERP를 커스터마이징하지 않고 표준적인 중소기업용 ERP를 사용하기 때문에 표준적인 데이터의 수집이 가능하다는 장점이 있다. 만일 본 연구를 프로세스 마이닝 도구의 기능으로 추가하여 중소기업용 ERP 솔루션에 적용한다면 중소기업에서는 추가적인 도움 없이도 자체적인 혁신 활동이 가능하여 기업 경쟁력 향상에 많은 도움을 받을 수 있을 것으로 기대된다.

머신러닝, 딥러닝과 같은 인공지능 기술은 주로 이미지 인식, 결과 예측 위주로 이용되고 있으며, 설명 가능한 인공지능 모델은 예측보다는 이미지 인식 등에서 인공지능이 해당 객체를 파악하는 과정을 사람에게 설명하기 위한 용도로 주로 활용되었다. 그러나 이를 프로세스 마이닝처럼 결과 예측보다는 원인분석이 더 중요한 업무에 활용한다면 원인분석 자동화의 용도로 확장이 가능하다는 것이 증명되었다. 예를 들어 보안 업계에서 활용되고 있는 인공지능을 이용한 침입 탐지 자동화의 경우, 단순히 자동 탐지한 결과를 확인하고 정확도만 계산하는 것이 아니라 일정 수준의 정확도가 확보되었다면 이를 설명 가능한 인공지능을 이용해 해당 알고리즘이 어떤 기준으로 침입을 탐지하는지 확인할 수 있으며 이를 기반으로 추가적인 보안 프로그램의 개선으로까지

지 이어질 수 있을 것이다.

본 논문에서는 3가지 실험 데이터에 대해 각각 다른 전처리 방법을 사용하였으며, 예측 모델에는 딥러닝 기반의 LSTM 외에도 Decision Tree, Random Forest, Multinomial Naïve Bayes, Logistic Regression 등을 실험하였고, 설명 가능한 인공지능 알고리즘은 LIME, SHAP, LRP 등을 적용하였다. 또한, 실험 결과에서는 각각의 예측 모델과 설명 가능한 알고리즘 간의 조합에 의해 확인된 성능의 차이와 장단점에 대해 세세히 설명하였다. 또한, 논문에서 사용된 모든 소스 코드와 가공된 데이터 역시 GitHub을 통해 공개하였다. 이를 통해 향후 프로세스 원인분석 자동화 연구를 개선하거나 보안, 금융 등 다른 인공지능 모델에 확장하여 실험하고자 하는 연구자들에게 참조할 수 있는 자료를 제공할 수 있을 것이다.

### 제 3 절 연구의 한계 및 향후 계획

본 연구에서는 3가지 다른 유형의 프로세스 마이닝 데이터와 LIME, SHAP, LRP의 3가지 설명 가능한 인공지능 알고리즘을 적용하여 장단점과 가능성을 확인하였다. 그러나 아직까지 설명 가능한 인공지능과 관련된 여러 연구가 진행 중인 상태로 안정성이나 구현 방법 등에 여러 가지 제약이 있었다. Gunning (2017)에 따르면 조만간 DARPA에서 구체적인 솔루션이 발표될 예정이며 국내의 Open XAI 프로젝트에서도 추가적인 알고리즘이 발표될 예정이다. 향후 설명 가능한 인공지능 알고리즘과 사용 가능한 도구의 발전에 따라 원인분석 알고리즘 역시 지속적인 업데이트가 필요할 것이다.

또한, 분석에 사용된 알고리즘과 데이터 전처리 방법들 역시 프로세스 마이닝 데이터에 따라 수행 방법이 표준화되기 힘들어 일반 기업에서 활용하기에는 어려운 점이 있다. 앞으로는 분석 대상이 되는 데이터를 입력하면 관련 알고리즘과 전처리 방법 등을 선택할 수 있고 그래픽 기반 사용자 인터페이스(GUI)를 통해 손쉽게 실행할 수 있도록 체계화된 솔루션을 개발하고자 한다.

본 연구에서는 기본적인 가정으로 프로세스 마이닝 프로젝트를 통해 도출된 전문가의 원인분석 결과가 정확할 것이라고 전제하였다. 그러나 사람의 직관이 관여된 결과가 더 정확하다고는 할 수는 없으므로 향후 실제 프로세스 마이닝 프로젝트에서 전문가가 업무 경험과 인터뷰 등을 통해 분석한 원인과 ‘설명 가능한 인공지능 기반의 이슈 원인분석’ 알고리즘만으로 자동으로 도출된 이슈 원인을 서로 비교하여 어느 쪽이 더 빠르고 정확한 결과를 제시하는지 비교해보는 연구도 필요할 것이다.

마지막으로 연구에 사용된 원인분석 자동화 기법을 프로세스 마이닝뿐만 아니라 침입 탐지 자동화 모델, 금융 시계열 분석 모델 등 다양한 분야의 인공지능 모델로 적용 범위를 넓히기 위한 연구를 진행할 예정이다.

## 참 고 문 헌

### 1. 국내문헌

- 강영식, & 이보경. (2016). *경영자와 실무전문가를 위한 프로세스 마이닝*. 서울: 한나래아카데미
- 강영식, 이현우, & 김병수. (2018). 프로세스 마이닝과 딥러닝을 활용한 구매 프로세스의 적기 입고 예측에 관한 연구. *Information Systems Review*, 20(4), 25-41.
- 권혁진. (2018). 프로세스 마이닝 기법을 활용한 모바일 게임 이용자 여정 분석 방안 연구. (박사학위논문, 숭실대학교 대학원).
- 김정연, 윤석준, & 이보경. (2018). 딥러닝 기반의 프로세스 예측에 관한 연구: 동적 순환신경망을 중심으로. *정보시스템연구*, 27(4), 115-128.
- 정찬일, & 이후진. (2019). Process Mining 기반의 PI 수행 기법에 관한 연구. *한국정보기술학회논문지*, 17(1), 1-9.
- 정찬일, & 이후진. (2019). 설명 가능한 인공지능 기반의 프로세스 마이닝 분석 자동화 연구. *전자공학회논문지*, 56(11), 45-51.
- 정찬일, & 이후진. (2020). 프로세스 분석을 위한 설명 가능한 인공지능 기법 비교 연구. *전자공학회논문지*, 57(8), 51-59.
- 송민석. (2013.11.20). 업무처리 시스템 분석하는 프로세스 마이닝, 빅데이터 시대의 핵심 역량으로. *포항공대신문*. Retrieved from <http://times.postech.ac.kr/news/articleView.html?idxno=7312>
- Loen Entertainment, & PMIG. (2017). 가입자 유형 분류와 타깃마케팅 방안 수립 사례 (멜론)[Blog]. Retrieved from <http://www.pmig.co.kr/case-melon-2>

## 2. 국외문헌

- Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, *10*(7), e0130140.
- Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K. R. (2010). How to explain individual classification decisions. *The Journal of Machine Learning Research*, *11*, 1803–1831.
- Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., ... & Zhang, Y. (2018). AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias.
- Berkes, P., & Wiskott, L. (2006). On the analysis and interpretation of inhomogeneous quadratic forms as receptive fields. *Neural computation*, *18*(8), 1868–1895.
- Blevi, L., Delporte, L., & Robbrecht, J. (2017). Process mining on the loan application process of a Dutch Financial Institute. *BPI Challenge*, 328–343.
- Bourne, M., Franco, M., & Wilkes, J. (2003). Corporate performance management. *Measuring Business Excellence*.
- Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.
- Caruana, R., & Niculescu-Mizil, A. (2006, June). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning* (pp. 161–168).
- Ceci, M., Lanotte, P. F., Fumarola, F., Cavallo, D. P., & Malerba, D. (2014, October). Completion time and next activity prediction

- of processes using sequential pattern mining. In *International Conference on Discovery Science* (pp. 49–61). Springer, Cham.
- Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2), 215–232.
- Cozijnsen, A. J., Vrakking, W. J., & van IJzerloo, M. (2000). Success and failure of 50 innovation projects in Dutch companies. *European Journal of Innovation Management*.
- Dahlgaard, J. J., Khanji, G. K., & Kristensen, K. (2008). *Fundamentals of total quality management*. Routledge.
- Davenport, T. H. (1993). *Process innovation: reengineering work through information technology*. Harvard Business Press.
- Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)* (pp. 1597–1600). IEEE.
- Drucker, P. F. (1998). The discipline of innovation. *Harvard business review*, 76(6), 149–157.
- Dupond, S. (2019). A thorough review on the current advance of neural network structures. *Annual Reviews in Control*, 14, 200–230.
- Erhan, D., Bengio, Y., Courville, A., & Vincent, P. (2010). Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3), 1.
- Evermann, J., Rehse, J. R., & Fettke, P. (2016, September). A deep learning approach for predicting process behaviour at runtime. In *International Conference on Business Process Management* (pp. 327–338). Springer, Cham.

- Gers, F. A., Schmidhuber, J., & Cummins, F. (1999). Learning to forget: Continual prediction with LSTM.
- Gevrey, M., Dimopoulos, I., & Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological modelling*, 160(3), 249–264.
- Giurgiu, I., & Schumann, A. (2019). Explainable failure predictions with rnn classifiers based on time series data. *arXiv preprint arXiv:1901.08554*.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial networks. *arXiv preprint arXiv:1406.2661*.
- Gunning, D. (2017). Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA), nd Web*, 2(2).
- Hammer, M. (1990). Reengineering work: don't automate, obliterate. *Harvard business review*, 68(4), 104–112.
- Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In *Neural networks: Tricks of the trade* (pp. 599–619). Springer, Berlin, Heidelberg.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Jung, C. & Lee, H. (2020). XAI-based automation for customer journey analysis, In *International Conference on Green and Human Information Technology*, Hanoi.
- Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., ... & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, 7(6),

673–679.

- La Rosa, M., & Soffer, P. (Eds.). (2013). *Business Process Management Workshops: BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012, Revised Papers* (Vol. 132). Springer.
- Landecker, W., Thomure, M. D., Bettencourt, L. M., Mitchell, M., Kenyon, G. T., & Brumby, S. P. (2013, April). Interpreting individual classifications of hierarchical networks. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)* (pp. 32–38). IEEE.
- Lison, P. (2015). An introduction to machine learning. *Language Technology Group (LTG)*, 1(35).
- Lopes, I. F., & Ferreira, D. R. (2019, September). A survey of process mining competitions: the BPI challenges 2011–2018. In *International Conference on Business Process Management* (pp. 263–274). Springer, Cham.
- Lundberg, S. M., Erion, G. G., & Lee, S. I. (2018). Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*.
- Lundberg, S., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.
- Montavon, G., Binder, A., Lapuschkin, S., Samek, W., & Müller, K. R. (2019). Layer-wise relevance propagation: an overview. *Explainable AI: interpreting, explaining and visualizing deep learning*, 193–209.
- Montavon, G., Lapuschkin, S., Binder, A., Samek, W., & Müller, K. R. (2017). Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65, 211–222.
- Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for

- interpreting and understanding deep neural networks. *Digital Signal Processing*, 73, 1–15.
- Nguyen, A., Dosovitskiy, A., Yosinski, J., Brox, T., & Clune, J. (2016). Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *arXiv preprint arXiv:1605.09304*.
- Poulin, B., Eisner, R., Szafron, D., Lu, P., Greiner, R., Wishart, D. S., ... & Anvik, J. (2006, July). Visual explanation of evidence with additive classifiers. In *Proceedings Of The National Conference On Artificial Intelligence* (Vol. 21, No. 2, p. 1822). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
- Povalyaeva, E., Khamitov, I., & Fomenko, A. (2017). Density Analysis of the Interaction With Clients. *BPI Challenge*
- Pyzdek, T., & Keller, P. (2014). Six sigma handbook. McGraw–Hill Education.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). “Why should I trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).
- Robson, G. D. (2010). *Continuous process improvement*. Simon and Schuster.
- Rodrigues, A. M. B., Almeida, C. F. P., Saraiva, D. D. G., Moreira, F. B., Spyrides, G. M., Varela, G., ... & Lopes, H. (2017). Stairway to value: mining a loan application process. *BPI Challenge*.
- Rokach, L., & Maimon, O. (2005). Top–down induction of decision trees classifiers—a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*,

35(4), 476–487.

- Rozinat, A. (2012.11.6). Case study : Process mining to improve a service refund process[Blog]. Retrieved from <https://fluxicon.com/blog/2012/11/case-study-process-mining-to-improve-a-service-refund-process>.
- Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *arXiv preprint arXiv:1708.08296*.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.
- Theis, J., & Darabi, H. (2019). Decay replay mining to predict next process events. *IEEE Access*, 7, 119787–119803.
- Van Der Aalst, W. M. P. (2011). *Process Mining: Discovery, Conformance and Enhancement*. Springer
- Van Der Aalst, W. M. P., Adriansyah, A., De Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., ... & Wynn, M. (2011, August). Process mining manifesto. In *International Conference on Business Process Management* (pp. 169–194). Springer, Berlin, Heidelberg.
- Van Der Aalst, W. M. P., Schonenberg, M. H., & Song, M. (2011). Time prediction based on process mining. *Information systems*, 36(2), 450–475.
- Van Dongen, B. (2012). *BPI Challenge 2012* (Version 1). 4TU.ResearchData. <https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f> ([ 'http://www.win.tue.nl/bpi/2012/challenge', 'https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b', 'https://doi.org/10.4121/uuid:7f787965-da13-4bb8-a3fd-242f08aef9c4'] )

- Van Dongen, B. (2017). *BPI Challenge 2017* (Version 1). 4TU.ResearchData. <https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b> ([<https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>])
- Van Oord, A., Kalchbrenner, N., & Kavukcuoglu, K. (2016, June). Pixel recurrent neural networks. In *International Conference on Machine Learning* (pp. 1747–1756). PMLR.

# ABSTRACT

## Study of Process Analysis Automation based on XAI

Jung, Chanyil

Major in Smart Convergence Product

Dept. of Smart Convergence Consulting

The Graduate School

Hansung University

The fields such as recognition and prediction that were considered what only humans could do are being automated with the recent development of artificial intelligence. However, human intervention is still regarded as essential in many areas of work. The process innovation project, one of the methodologies for business management, is one of the areas that was considered impossible without an expert's judgment process. Business process innovation is divided into the steps of deriving an issue through status analysis and establishing an to-be process model by analyzing the cause of the issue. The process mining contributed significantly to effectively conducting the status analysis by automatically discovering the as-is process model using an information system log. However, it is still recognized that the experience and intuition of experts are very

important to understand the root cause of the issue.

In this study, based on the data of process mining, whether the result of the business process is normal or whether an issue will occur is predicted in advance using a machine learning model. Also, the cause of the issue was automatically found by analyzing the XAI (eXplainable Artificial Intelligence) algorithm that can explain the prediction model. By automatically deriving the causes of business process issues in this way, the scope of the use of artificial intelligence has been extended to the area of issue cause analysis for business process innovation that was previously considered impossible without human intervention. Furthermore, business process innovation projects that the companies operated with significant time and money can be conducted with far less time and human resources that can contribute to the company's sustainable innovation.

The research was conducted on the results of process mining and system log data published by banks, electronics manufacturing companies, and internet service companies. First, it proved that it is possible to automate cause analysis through an algorithm by analyzing the cause of the cancellation of the customer's loan application based on the bank's loan process. Additionally, for the refund process of manufacturing companies, various machine learning algorithms including deep learning and actively researched algorithms in the field of XAI were applied to analyze and compare. A deep learning model increases the accuracy of the prediction than a traditional machine learning model. On the other hand, the traditional machine learning model is advantageous for the cause analysis that they are in the trade-off relationship. However, the most important factor that affects the results is data pre-processing and it was confirmed that the performance of the cause analysis algorithm increased as the process flow, execution time, and additional attributes

were added to the process log. Based on the result, a process model is created based on the user log of the mobile app of an music streaming service company. Then the machine learning model is applied to predict the user's ticket payment failure in advance and automatically derived the cause of the payment failure through an XAI algorithm. The automatically derived cause of the failure was the same as the cause of failure previously analyzed by process mining experts, and through this, it was proved that the automated issue cause analysis algorithm has excellent performance to replace the analysis of experts.

The 'eXplainable Artificial Intelligence-based Process Analysis Automation' presented in the study extended the scope of application of the artificial intelligence algorithm to the field of issue cause analysis that were considered impossible without a human's judgment process. In addition, the usefulness of the current process mining tool was improved by automation of cause analysis. With this, in the modern business environment where competition among industries is fierce and rapidly changing, process innovation activities that are essential for the survival of companies can be performed with much less time and human resources. Moreover, by allowing the process innovation system through automation, it will be of practical help in strengthening the competitiveness and continuous growth of the company.

In the future, the development will progress by standardizing the data pre-processing and applied algorithms that they can be added as the major function of the process mining solution while researching to increase reliability by applying the 'eXplainable Artificial Intelligence-based Process Issue Cause Analysis Automation' technique to various business process data. In addition, the application of the artificial intelligence-based cause analysis technique that explains not only process mining but also numerous tasks that require cause analysis will provide

various academic and economic effects.

【Keywords】 Explainable Artificial Intelligence, Process Mining,  
Causal Analysis Automation