J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

액티비티별 특징 정규화를 적용한 LSTM 기반 비즈니스 프로세스 잔여시간 예측 모델

LSTM-based Business Process Remaining Time Prediction Model Featured in Activity-centric Normalization Techniques

함 성 훈[1] 안 현[1] 김 광 훈[1*]
Seong-Hun Ham Hyun Ahn Kwanghoon Pio Kim

요 약

최근에 많은 기업 및 조직들이 비즈니스 프로세스 모델의 효율적 운용을 위해 예측적 프로세스 모니터링에 관심이 높아지고 있다. 기존의 프로세스 모니터링은 특정 프로세스 인스턴스의 경과된 실행상태에 초점을 두었다. 반면, 예측적 프로세스 모니터링은 특정 프로세스 인스턴스의 미래의 실행상태에 대한 예측에 초점을 둔다. 본 논문에서는 예측적 프로세스 모니터링 기능 중 하나인 비즈니스 프로세스 인스턴스 실행 잔여시간 예측기능을 구현한다. 잔여시간을 효과적으로 모델링하기 위해 액티비티별 속성에 따른 시간특징 값 분포 차이를 고려하여 액티비티별 특징 정규화를 제안하고 예측모델에 적용한다. 본 논문에서 제안된 모델의 예측성능 우수성을 입증하기 위해서 4TU.Centre for Research Data에서 제공하는 실제 기업의 이벤트 로그 데이터를 통해 선행연구들과 비교평가 한다.

☞ 주제어 : 예측적 프로세스 모니터링, 잔여시간 예측, LSTM 모델, 딥러닝, 프로세스 마이닝

ABSTRACT

Recently, many companies and organizations are interested in predictive process monitoring for the efficient operation of business process models. Traditional process monitoring focused on the elapsed execution state of a particular process instance. On the other hand, predictive process monitoring focuses on predicting the future execution status of a particular process instance. In this paper, we implement the function of the business process remaining time prediction, which is one of the predictive process monitoring functions. In order to effectively model the remaining time, normalization by activity is proposed and applied to the predictive model by taking into account the difference in the distribution of time feature values according to the properties of each activity. In order to demonstrate the superiority of the predictive performance of the proposed model in this paper, it is compared with previous studies through event log data of actual companies provided by 4TU.Centre for Research Data.

☞ keyword : predictive process monitoring, remaining time prediction, LSTM model, deep learning, process mining

1. 서 론

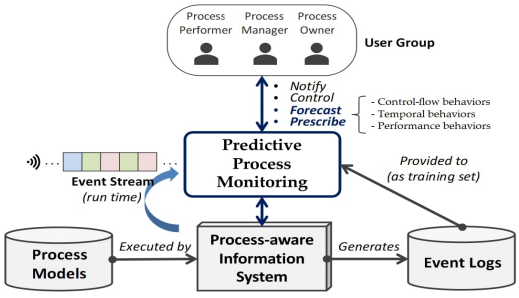
최근에 많은 조직 및 기업들이 프로세스 기반 정보시스템(Process-aware Information System[1], 이하 PAIS)을 기반으로 비즈니스 프로세스를 관리한다. PAIS는 비즈니스 프로세스 실행을 자동화하고 모니터링 및 분석 기능을 통해 효율적인 프로세스 관리 및 운영을 지원한다. 특히, 모니터링 기능은 프로세스 현황정보를 실시간으로 수집해 요약 및 시각화한다. 이를 기반으로 프로세스 실행을 효과적으로 제어하거나 관련된 의사결정 활동이 가능하다. 비즈니스 프로세스 이벤트 로그(이하 로그)는 프로세스 모니터링의 원천 데이터이다. 로그는 프로세스 인스턴스의 수행과정에서 발생한 모든 실행이력 정보를 포함한다. 최근에는 이러한 로그에 프로세스 마이닝[2,3] 기법들을 적용하여 조직에서 실제 실행되고 있는 비즈니스 프로세스 모델을 발견하는 연구가 수행되었다. 또한 발견

[1] Computer Science, Kyonggi University, Suwon, 16227, South Korea
[Received 17 April 2020, Reviewed 23 April 2020, Accepted 20 May 2020]
* Corresponding author (kwang@kgu.ac.kr)
☆ 이 논문은 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임. (중견연구자지원사업 No. 2017R1A2B2010697)
☆ 본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행되었음. (과제번호: NRF-2018R1C1B5086414)
☆ 본 연구는 경기대학교 일반대학원 연구원장학생프로그램의 지원을 받아 수행되었음

한 모델을 기반으로 프로세스 실행의 문제점을 파악하고 개선에 활용하기 위한 연구 및 적용사례들이 증가하고 있다.

한편, 대규모 데이터 수집 및 처리에 대한 컴퓨팅 성능의 발전으로 딥러닝 기반의 지능형 예측 기술이 다양한 비즈니스 문제 해결에 적용되고 있다. 이와 관련하여, 기존의 프로세스 모니터링 기술에서 확장된 예측적 프로세스 모니터링(predictive process monitoring[4])에 대한 연구가 활발히 진행되고 있다. 기존의 프로세스 모니터링 및 프로세스 마이닝 기술이 프로세스 인스턴스들의 과거 수행이력 및 현황 정보를 제공했다면, 예측적 프로세스 모니터링은 수행중인 프로세스 인스턴스들의 미래를 예측하는 것에 초점을 맞춘다. 예측적 프로세스 모니터링의 연구 주제로는 프로세스 인스턴스의 다음 액티비티(업무 태스크) 및 실행경로 예측[4], 인스턴스의 잔여시간 예측[4,5] 설비의 고장 예측[6] 등이 있다. 이를 위해서, 각 예측 문제 별로 예측모델을 개발해야한다. 최근에는 로그를 훈련 데이터로 사용하는 딥러닝 기반의 예측 모델들이 연구되고 있다. 본 논문에서는 프로세스 인스턴스 잔여시간 예측을 위한 LSTM(Long Short-Term Memory) 신경망 모델을 개발하고 예측성능평가를 수행한다. 기존의 연구들이 특징설계에 중점을 두고 잔여시간 예측 정확도를 높이고자 했다면, 본 논문에서는 데이터 전처리 부분에 초점을 맞춘다.

논문의 구성은 다음과 같다. 2장에서는 프로세스 인스턴스 잔여시간 예측 관련 선행연구들에 대해 설명하며, 3장에서는 이론적 배경인 잔여시간의 정의와 LSTM 신경망에 대해 설명한다. 4장에서는 특징 설계와 제안하는 액티비티별 특징 정규화를 설명한다. 5장에서는 제안된 방법과 관련연구[4,5]에 대해서 예측성능평가 및 비교하며, 마지막으로 6장에서는 결론을 제시한다.



(그림 1) 예측적 프로세스 모니터링의 개념
(Figure 1) The Predictive Process Monitoring Concept

2. 관련 연구

프로세스 인스턴스 잔여시간(이하 잔여시간) 예측 관련 초기 연구로써, van der Aalst et al.[7]은 잔여시간의 개념을 정형적으로 정의하고 비모수적(nonparametic) 회귀기법을 통해 이를 예측하는 방법을 제안했다. 이후에 잔여시간 예측의 정확도를 높이기 위해 다양한 연구들이 선행되었다. Polato et al.[8]는 오토마타 기반 전이시스템(transition system)을 통해 잔여시간을 예측하고자 하였고, Rogge et al.[9]는 stochastic Petri net을 사용하여 프로세스의 실행경로를 모델링하고 이로부터 잔여시간을 예측하는 기법을 제안하였다. 이와 비슷한 연구로서, Verenich et al.[10]는 BPMN(Business Process Model and Notation [11])으로 표현된 프로세스의 제어흐름 분석을 통해 예측 정확도를 높이고자 하였다. 위의 선행연구들의 경우, 프로세스의 제어흐름을 다양한 수학적 모델로 나타내고, 이를 기반으로 잔여시간 예측을 수행한 연구들이다.

한편, LSTM 신경망 기반의 잔여시간 예측 연구들이 수행되었다. LSTM 신경망은 순환신경망 계열로, 제어흐름에 의해 실행되는 액티비티간의 인과관계를 모델링하는데 효과적이다. Tax et al.[4]은 잔여경로를 예측하고 예측된 경로의 잔여시간을 예측하는 연구를 수행했다. 또한 비업무시간에 의한 잔여시간의 증가를 모델링하기 위해 시간 및 요일 값을 특징으로 사용했다. Navarin et al.[5]은 입력 값을 통해 즉시 잔여경로를 예측했다. 또한, 수행자별 업무숙련도에 따른 업무수행시간을 모델링하기 위해서 액티비티 수행자 정보를 특징으로 사용했다. Verenich et al.[12]는 수학적 모델기반 접근법[7-11]과 LSTM 신경망 모델기반 접근법[4,5]들의 잔여시간 예측 연구를 비교 및 분석하였다. 비교실험 결과로, 대다수의 잔여시간 예측 실험에서 LSTM 신경망 모델기반의 접근방법들이 우수한 성능을 보였다. 본 논문에서는 기존의 LSTM 신경망 모델기반의 예측 모델[4,5]와 비교하여 개선된 정규화 방법을 통해 예측성능을 높이고자한다.

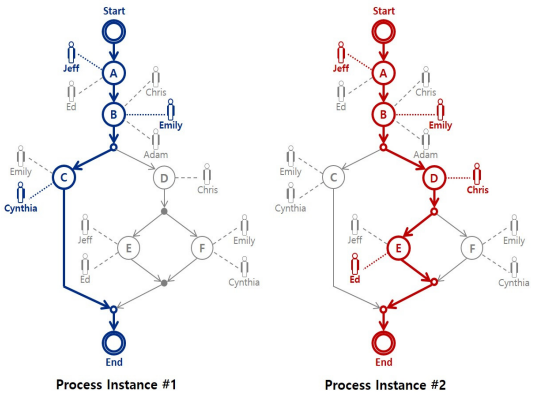
3. 이론적 배경

3.1 프로세스 인스턴스 및 잔여시간 개념

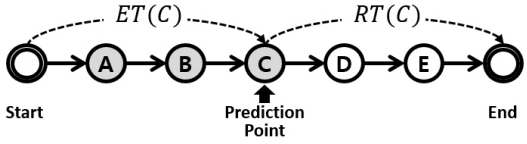
프로세스 인스턴스(이하 인스턴스)는 개별적인 비즈니스 프로세스 실행 건이다. 그림 2와 같이 각각의 인스턴스는 서로 다른 실행경로와 업무수행자에 의해 수행될 수 있다. 인스턴스의 시작에서 종료까지의 모든 수행 과

정들은 순차적으로 로그에 저장된다. 개별 인스턴스에 해당되는 이벤트들의 집합을 트레이스(trace)라고 정의한다.

트레이스는 액티비티 또는 업무수행자 중심으로 나타낼 수 있다. 액티비티 트레이스의 경우(예: <A-B-C>), 프로세스 모델 발견[13] 및 제어흐름 분석[14-16]의 기본 데이터로 활용된다. 업무수행자 트레이스의 경우(예: <Jeff-Emily-Cynthia>), 인적자원 중심의 비즈니스 프로세스 분석[17-22]에 활용된다. 이와 같이, 트레이스는 프로세스 분석의 기본 데이터 블록이다. 예측적 프로세스 모니터링도 트레이스를 중심으로 특징을 추출하고 이로부터 예측모델을 생성한다.



(그림 2) 프로세스 인스턴스 예제
(Figure 2) Process Instance Examples

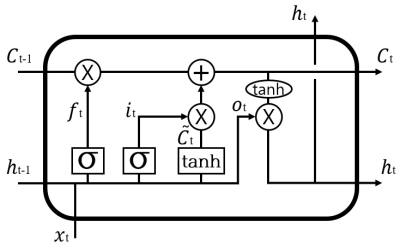


(그림 3) 프로세스 인스턴스 잔여시간 및 경과시간
(Figure 3) The Remaining Time and the Elapsed Time on a Business Process Instance

본 논문의 핵심 개념인 잔여시간의 경우, 인스턴스와 각 액티비티에 해당되는 이벤트들의 타임스탬프로 모델링된다. 그림 3은 액티비티(α_A~α_E)로 표현된 프로세스 인스턴스와 관련 시간속성에 대한 개념을 나타낸다. 인스턴스의 첫 번째 이벤트(α_A)와 마지막 이벤트(α_E)의 타임스탬프는 각각 인스턴스의 시작시간 및 종료시간이며, 예측 대상인 특정 이벤트(α_C)의 잔여시간($RT(\alpha_C)$)은 α_C와 마지막 이벤트 α_E과의 타임스탬프 차이로 정의된다. 반면에, 특정 이벤트의 경과시간($ET(\alpha_C)$)은 α_C와 첫 번째 이벤트 α_A와의 타임스탬프 차이로 정의되며, 이를 잔여시간 예측을 위한 입력특징으로 사용한다.

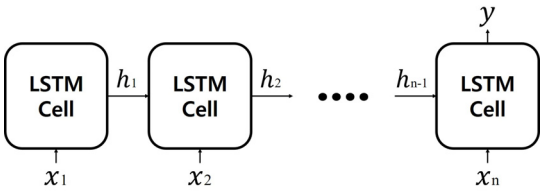
3.2 LSTM 신경망

본 논문에서는 프로세스 인스턴스의 잔여시간 예측을 위해 LSTM 신경망[23]을 사용한다. LSTM 신경망은 기존의 RNN(Recurrent Neural Network)의 기울기 소실 문제를 개선한 신경망으로 망각 게이트(forget gate)와 입력 게이트(input gate)를 통해 과거정보와 현재정보의 기억유무를 계산해 효율적으로 다음 셀에 정보를 전달한다. 그림 4는 LSTM 셀의 구조를 나타낸다.



(그림 4) LSTM 셀 구조
(Figure 4) The LSTM Cell Structure

인스턴스를 입력받아 잔여시간을 예측하기 위해서 다대일(many-to-one) 구조의 LSTM 신경망 모델을 사용한다. 인스턴스 내부에 있는 다수의 액티비티를 순차적으로 입력하여 하나의 결과를 얻을 수 있는 해당 구조는 액티비티간의 인과관계를 모델링 하는데 효과적이다. 아래 그림 5는 LSTM 신경망 모델의 다대일 구조를 나타낸다.



(그림 5) LSTM 모델의 다대일 응용아키텍처 구조
(Figure 5) Many-to-One Application Architecture of LSTM Model

4. 예측모델 설계 및 액티비티별 특징 정규화

4.1 특징 설계

이벤트 로그는 인스턴스 아이디, 액티비티 아이디, 타임스탬프, 그리고 수행자 아이디를 기본적인 속성으로 가진다. 입력 특징 벡터를 생성하기 위해서 네 가지의 기본적 속성으로부터 특징을 추출한다.

- 액티비티 아이디(Activity ID, 이하 ID): 수행된 이벤트의 액티비티 아이디.
- 인스턴스 시작으로부터의 경과시간(Elapsed Time from Instance Start, 이하 ET_s): 인스턴스의 첫 이벤트와 현재 이벤트의 타임스탬프간의 시간 차이.
- 직전 이벤트로부터의 경과시간(Elapsed Time from Last Event, 이하 ET_e): 현재 이벤트와 직전이벤트 타임스탬프간의 시간차이.
- 당일 자정으로부터의 경과시간(Elapsed Time from Midnight, 이하 ET_m): 당일 자정과 현재 이벤트 타임스탬프간의 시간 차이. 해당 속성은 현재의 시간을 정량적으로 나타내는 속성이다.
- 요일정보(Day of the Week, 이하 DW): 현재 이벤트 타임스탬프의 요일 정보.
- 액티비티 수행자(Activity Performer, 이하 AP): 현재 액티비티를 수행한 수행자 정보.

ET_s, ET_e, ET_m, DW는 시간적 특징 값으로 정규화의 대상이다. ET_s, ET_e는 경과시간을 통해 인스턴스의 경과정도를 모델링한다. ET_m와 DW는 현재 인스턴스의 시간정보를 표현하여 이후 비업무시간에 따른 잔여시간 증가를 모델링한다. ID와 AP는 각 특성에 대한 고유의 아이디이다.

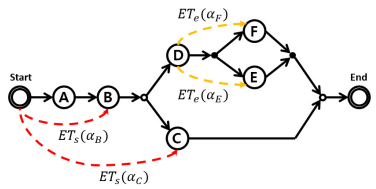
4.2 액티비티별 특징 정규화

정규화는 특징 간 값의 범위를 균일하게 만들어 모델 학습의 안전성을 향상시키는 기법이다. LSTM 신경망 기반의 선행연구인 Tax et al.[4]에서는 각 특징의 평균값(x_{mean})을 이용해 평균 정규화를 수행하였으며 Navarin et al.[5]은 각 특징의 최대값($x_{\max}$)과 최소값($x_{\min}$)을 이용해 Min-Max 정규화를 수행했다. 아래의 식은 두 정규화 식을 나타낸다.

$$x_{norm} = \frac{x}{x_{mean}} \qquad \text{<수식 1>}$$

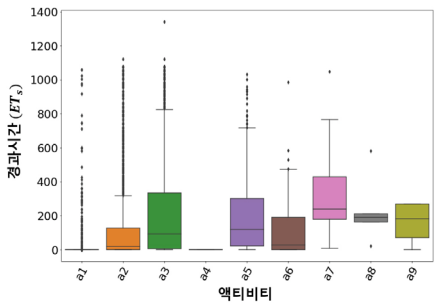
$$x_{norm} = \frac{x - x_{\max}}{x_{\max} - x_{\min}} \qquad \text{<수식 2>}$$

선행 연구들은 정규화를 통해서 특징 간 값의 범위를 균일하게 만들었다. 하지만, 일부 특징 내부에서는 액티비티별로 속성에 따라서 값의 분포가 다르다. 액티비티별 속성은 프로세스에서 액티비티의 위치와 액티비티의 업무처리시간 등이 있다. 자세한 사항은 그림 10을 통해 나타낸다. 액티비티 B는 항상 액티비티 C보다 먼저 실행된다. 즉, 액티비티 B의 ET_s는 액티비티 C의 ET_s보다 항상 값이 작다. ET_e의 경우 각 액티비티의 처리시간에 따라서 값의 분포가 결정된다. ET_m과 DW는 액티비티에 상관없이 고정적인 데이터 범위를 가진다.



(그림 6) 액티비티별 속성에 따른 시간특징값 분포 차이
(Figure 6) Difference in Distribution of Time Feature Values Aaccording to Properties by Activity

이러한 현상이 실제 데이터에도 나타나는지 확인하기 위해 helpdesk 로그에서 ET_s를 추출하고 액티비티별로 분류한다. 이는 그림 11에 boxplot으로 나타낸다.



(그림 7) 액티비티별 경과시간 분포
(Figure 7) Distribution of Elapsed Time by Activity

***Algorithm* : ET_s 특징 추출 및 액티비티별 특징 정규화**

Input: 이벤트 로그 (ε^{all})
Output: 정규화된 ET_s집합 (nET_s^{all})

```
1: struct{                                          *이벤트의 구조체
2:         p : process instance ID                  *프로세스 인스턴스 아이디
3:         α : activity ID                          *액티비티 아이디
4:         t : timestamp                            *타임스탬프(이벤트 종료시간)
5: } ε[N];                                          *N: 이벤트 로그의 개수
6: #Preprocessing
7: Λ = {I_0, I_1, ..., I_{g-1}}                     *인스턴스 집합
8: I_i = {ε_0, ε_1, ..., ε_{k-1}}, 0 ≤ i ≤ g − 1, ∀ε_j.p = I_i   *인스턴스(이벤트 집합)
9: A = {α_0, α_1, ..., α_{p-1}}                     *액티비티 집합
10: for ∀I_i in Λ
11:    for ∀ε_j in I_i
12:       ET_s(I, ε_j) = ε_j.t − ε_0.t              *해당 이벤트에 대한 ET_s 계산
13:       T_s^{ε_j.α} ← T_s^{ε_j.α} ∪ ET_s(I_i, ε_j)   *계산된 ET_s를 액티비티별 그룹에 저장
14: for ∀α_l in A
15:    α_l^{ET_s}.max = max(T_s^{α_l})              *각 액티비티별 최대값 계산
16:    α_l^{ET_s}.min = min(T_s^{α_l})              *각 액티비티별 최소값 계산
17: for ∀I_i in Λ
18:    for ∀ε_j in I_i
19:       nET_s^{all} ← nET_s^{all} ∪ (ET_s(I_i, ε_j) − ε_j.α^{ET_s}.min) / (ε_j.α^{ET_s}.max − ε_j.α^{ET_s}.min)
                                                    *이벤트의 액티비티 아이디 값을 통해
                                                     액티비티별 특징 정규화
```

(그림 8) 특징 추출 및 액티비티별 특징 정규화 알고리즘
(Figure 8) Algorithm of Normalization by Activity

본 논문에서는 액티비티별 시간특징 값의 차이를 고려하기 위해 액티비티별 특징 정규화를 제안한다. 적용되는 특징은 ET_s와 ET_e이다. 그림 12는 ET_s에 대한 특징 추출 및 액티비티별 특징 정규화 알고리즘이다. 이벤트 로그를 입력으로 받아 ET_s를 추출하고 액비티비별로 특징정규화 하여 반환한다. helpdesk 로그의 인스턴스 #117($\alpha_1-\alpha_2-\alpha_3$)를 통해 예를 든다. α_3과α_1의 타임스탬프 차이로 $ET_s(117,\alpha_3)$가 계산된다. 현재 액티비티는 α_3이므로 $\alpha_3^{ET_s}$의 최대값과 최소값을 통해 정규화 된다. 해당 알고리즘의 시간 복잡도는 $O(\iota_i\times\epsilon_j+\alpha_l\times T+\iota_i\times\epsilon_j)$이다. $\iota_i\times\epsilon_j$와 $\alpha_l\times T$는 모든 이벤트에 대한 전수탐색을 의미한다. 따라서 로그의 전체 이벤트 개수가 n일 경우, 시간 복잡도는 $O(n)$으로 표현된다. 기존의 일반적인 정규화(수식 1,2)의 시간복잡도 또한 $O(n)$으로 액티비티별 특징 정규화와 동일하다. ET_e의 경우 위 알고리즘의 10번 줄의 코드가 $ET_e(\iota_i\epsilon_j)=\epsilon_j.t-\epsilon_{j-1}.t.$으로 대체된다. ET_m과 DW는 Min-Max 정규화(수식 2)를 적용한다.

5. 실험 및 성능평가

본 장에서는 LSTM 기반의 잔여시간 선행연구들[4,5]과 본 논문에서 제안하는 모델을 이용하여 실험 및 성능평가를 수행한다. 실험 환경은 표 1과 같다.

(표 1) 실험 환경
(Table 1) Experimental Setup

실험환경	값
운영체제	Ubuntu 16.04 LTS
언어	python 3.7
라이브러리	pytorch 1.3, pandas, sklearn.metrics, Seaborn, datetime
그래픽 카드	GeForce 2080 ti

5.1 실험 데이터셋 및 비교 모델

본 실험에서 사용하는 데이터셋은 4TU.Centre for Research Data[24]에서 제공된 실제 이벤트 로그이다. 그 중에서 프로세스 마이닝 관련 대회인 BPIC(Business

Process Intelligence Challenge)에서 공개된 BPIC2012, 2017과 연구목적으로 공개된 HelpDesk 데이터셋을 잔여시간 예측 실험에 사용한다. 각 데이터셋의 세부사항은 다음과 같다.

- HelpDesk 데이터셋: 이탈리아 소프트웨어 기업의 헬프 데스크에서 제공하는 티켓팅 관리 프로세스의 이벤트 로그이다. 2010년부터 2014년까지 약 4년 동안 수집되었다. 해당 로그는 CSV(Comma-Separated Values) 파일 형식으로 저장되어 있다.
- BPIC2012 데이터셋: 글로벌 금융기관 Dutch Financial Institute의 개인 대출 및 당좌대월(overdraft) 신청 프로세스 로그이다. 로그는 2011년 10월부터 2012년 3월까지 수집됐다. 해당 프로세스는 업무수행자가 처리하는 일반 태스크(manual task)와 프로그램이 자동적으로 처리하는 자동 태스크(automatic task)를 포함한다. 본 논문에서는 필터링을 통해 일반 태스크 로그만을 사용하였다. 해당 로그는 XES(eXtensible Event Stream) 포맷[25]으로 저장되어 있다.
- BPIC2017 데이터셋: BPIC2012와 동일하게 Dutch Financial Institute의 대출신청 로그이다. 2016년부터 2017년 2월까지 수집되었다. 프로세스 지원 시스템의 변경으로 BPIC2012와는 다른 모델에서 실행되었다. 해당 로그는 XES 포맷으로 저장되어있다.

(표 2) 프로세스 이벤트 로그 세부 정보
(Table 2) Process Event Log Detail Information

로그	Helpdesk	BPIC2012	BPIC2017
패턴 수/인스턴스 수	155/3804	2264/9658	5624/31509
액티비티 개수	9	6	24
잔여시간 평균	211.1	273.6	525.5
인스턴스 평균 길이	3.6	7.5	15.5
반복실행 평균 횟수	1.1	6.8	1.7

표 2는 데이터셋에 대한 통계 정보를 나타낸다. 첫 항목인 '패턴 수/인스턴스 수'는 로그가 얼마나 다양한 패턴의 인스턴스를 보이는지 나타낸다. 다음 항목인 '액티비티 개수'는 로그에서 발견되는 액티비티의 종류이다. 나머지 항목은 잔여시간, 인스턴스 길이, 반복실행 횟수의 평균이다. 표 2에 따르면 helpdesk 는 액티비티 개수를 제외한 모든 항목에서 가장 낮은 수치를 보인다. BPIC2012는 가장 낮은 액티비티 개수를 가지지만 패턴의 비율과 반복실행 평균 횟수가 가장 높다. BPIC2017은 액티비티의 개수와 잔여시간 및 인스턴스 평균 길이가 가장 길다. 이러한 서로 다른 특성을 가지는 로그들을 사용해 잔여시간 예측 비교성능평가를 진행한다.

제안된 모델과 비교성능평가를 수행하는 LSTM 신경망 기반 선행연구들[4,5]의 세부사항 및 차이점은 아래 표에서 자세히 나타낸다. Tax et al.[4]은 다섯 가지 특징들(ID,ET_s,ET_e,ET_m,DW)을 사용하였다. 이를 통해 잔여경로 예측을 수행하고, 예측된 경로에 대해서 잔여시간을 예측했다. 정규화는 네가지 특징(ET_s,ET_e,ET_m,DW)에 대해서 평균정규화(수식 1)를 적용했다. Navarin et al.[5]은 특징 AP를 추가적으로 사용했다. 정규화는 네가지 특징(ET_s,ET_e,ET_m,DW)에 대해서 Min-Max 정규화(수식 2)를 적용했다. 제안된 논문은 Navarin과 동일한 특징을 사용하였으며 액티비티별로 분포 차이를 보이는 특징(ET_s,ET_e)에 대해서 액티비티별 특징 정규화(그림 12)를 적용하고 고정적인 범위를 보이는 특징(ET_m,DW)에 대해서는 Min-Max 정규화(수식 2)를 적용했다.

(표 3) 예측 모델별 세부사항
(Table 3) Predictive Models-Specific Details

Model	Tax[4]	Navarin[5]	Ours
입력 특징	ID,ET_s,ET_e, ET_m,DW	ID,ET_s,ET_e, ET_m,DW,AP	ID,ET_s,ET_e, ET_m,DW,AP
정규화	평균 정규화	Min-Max 정규화	액티비티별 특징 정규화, Min-Max 정규화
인코딩	One-hot 인코딩	One-hot 인코딩	One-hot 인코딩
추가 모델	잔여경로 예측 모델	-	-

5.2 예측모델 설정 및 성능지표

본 논문의 LSTM 모델의 매개변수 및 세부사항은 표 4와 같다.

(표 4) LSTM 모델 파라미터 세부사항
(Table 4) The Detailed Manifest of the LSTM Model Parameters

세부사항	비고
일괄처리량	1000
학습 반복 횟수	500
은닉계층 수	3
학습률	0.001
드롭아웃	0.1
손실 함수	평균절대오차
최적화 알고리즘	Adam optimizer

해당 파라미터는 세 가지 로그에 대해서 가장 높은 성능을 보이는 수치이다. 또한 과적합을 방지하기 위해서 학습조기종료를 사용한다. LSTM 신경망에 입력하기 위해 인스턴스들의 길이를 패딩을 적용해 고정했다. 학습 데이터는 훈련 집합 50%, 검증 집합 20%, 그리고 테스트 집합 30%로 구성된다. 또한, 예측 모델의 성능평가기준으로서 평균절대오차(Mean Absolute Error, 이하 MAE) 손실함수를 사용한다. MAE 값은 모든 관측값(y_j)과 모델의 예측값($\hat{y}_j$) 간 절대 차이값의 합을 데이터 수(n)로 나눈 값이다. 예측 대상인 잔여시간은 통합 단위로 일(day)를 사용하며 소수점 4번째 자리에서 반올림 한다. 아래의 수식은 평균절대 오차를 나타낸다.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \qquad \text{<수식 3>}$$

5.3 실험 및 결과

본 실험에서는 잔여시간 예측 선행연구들[4,5]와 제안된 예측모델간의 예측성능평가 및 비교를 수행한다. 세부적으로 두 가지의 실험이 수행된다. 첫 번째는 서로 다른 특성을 가진 로그들(Helpdesk, BPIC2012, BPIC2017)을 통해 프로세스에 따른 각 모델의 예측성능을 평가한다. 두 번째는 전처리를 통해 단순화된 BPIC2012 로그에 대하여 각 모델의 예측성능을 평가한다. 두 번째는 전처리를 통해 단순화된 BPIC2012 로그에 대하여 각 모델의 예측성능을 평가한다. 실험결과로 나오는 MAE 오차는 실제 시간의 차이를 의미한다. MAE 1은 실제 시간 1일을 나타내며 0.5의 경우 12시간을 의미한다.

표 5는 세 로그에 대한 잔여시간 예측 결과를 나타낸다. Prefix는 이전에 실행된 이벤트의 최소 개수를 나타낸다. 모든 데이터셋에서 제안된 모델은 가장 높은 성능을 보였고 Navarin 모델은 두 번째로 높은 성능을 보였다. 두 모델의 예측성능 차이는 Helpdesk에서 1일, BPIC2012에서 3일 7시간, 그리고 BPIC2017에서 5일 12시간이다. 두 모델이 세 로그에 대해서 보이는 예측성능 차이는 이러한 로그의 특성을 통해서 분석된다. 잔여시간 평균과 인스턴스 평균 길이는 프로세스 모델의 규모를 의미한다. 패턴비율은 해당 프로세스 제어흐름의 복잡도를 의미한다. 단, 반복적 제어흐름에서는 다른 제어흐름에 비해 많은 패턴이 발생 할 수 있다. 액티비티 개수가 24개, 잔여시간 평균이 525일, 그리고 인스턴스 평균 길이가 15이다. 이는 나머지 두 로그보다 2배이상 높은 수치이다. BPIC2012는 패턴비율이 24%이며 반복실행 평균횟수가 6.8로 BPIC2017보다 더 높다. 하지만 액티비티의 개수가 6개이며 로그에 포함된 반복실행이 모두 하나의 액티비티가 반복실행 하는 단순반복실행이다. Helpdesk는 가장 적은 패턴수와 인스턴스 길이를 가진다. 따라서 BPIC2017, BPIC2012, 그리고 Helpdesk 순으로 프로세스가 복잡한 것으로 분석된다. 위 결과에 따라서 제안된 모델은 Navarin의 모델보다 예측성능이 높다. 특히, 복잡한 프로세스에서 예측성능의 차이는 더욱 증가한다.

두 번째 실험은 첫 번째 실험과는 반대로 단순한 로그에 대한 예측성능 비교이며 결과는 표 6에 나타난다. 가장 적은 액티비티를 가지며 모든 반복제어흐름이 단순반복인 BPIC2012에 전처리를 하여 단순한 로그를 생성한다. 두 가지 전처리를 통해 각각 로그를 생성한다. 먼저 잔여시간 이상치를 가지는 인스턴스들을 제거한 로그를 생성한다. 두 번째로는 폴딩기법으로 반복적 제어흐름을 제거한 로그를 생성한다. 폴딩기법은 동일한 액티비티가 반복실행 되는 경우 하나로 그룹핑 하는 기법이다. 이상치가 제거된 BPIC2012에서는 모든 모델의 예측성능이 향상되었으며 제안된 모델이 가장 높은 성능을 보였다. 반복적 제어흐름을 제거한 BPIC2012에서는 Tax모델이 가장 높은 성능을 보였으며 제안된 모델은 약 43분 차이로 두 번째의 성능을 보였다. 따라서 단순한 프로세스에서도 제안된 모델이 높거나 근사한 성능을 보이는 것을 확인했다.

(표 5) 비즈니스 프로세스 인스턴스 잔여시간 실험 결과
(Table 5) Experimental Results on the Remaining Times of Business Process Instances

Model error	Model	HelpDesk			BPIC2012			BPIC2017		
		prefix 0	prefix 1	prefix 2	prefix 0	prefix 2	prefix 4	prefix 0	prefix 2	prefix 4
MAE (day)	Tax[4]	6.4343	5.4374	4.7786	71.219	69.7228	65.9011	60.3973	55.8966	49.9752
	Navarin[5]	5.8152	4.7826	3.8145	9.9319	9.7289	9.4520	12.1006	12.0012	11.7875
	Ours	**4.8397**	**3.7168**	**2.452**	**6.6585**	**6.3203**	**5.8352**	**6.6469**	**6.2125**	**5.6491**

(표 6) 반복제어흐름 및 이상치 실험 결과
(Table 6) Experimental Results on Repetition Control Flow and Outlier

Model	MAE (BPIC 2012)		
	Original	Without-Outliers	Loop-Body only (without repetition)
Tax[4]	71.219	35.4272	**5.2735**
Navarin[5]	9.9319	9.6185	8.0434
Our Model	**6.6585**	**6.2918**	5.3032

6. 결 론

본 논문에서는 액티비티별 시간특징 값 분포 차이를 고려하기 위해 액티비티별 특징 정규화를 제안하고 모델에 적용했다. 제안된 모델의 예측 성능 우수성을 입증하기 위해 4TU.Centre for Research Data[26]에서 제공된 실제 기업의 로그를 통해 실험을 수행했다. 첫 번째 실험은 세 가지 로그에 대한 잔여시간 예측이 수행되었으며 제안된 모델은 모든 로그에서 가장 높은 예측성능을 보였다. 프로세스의 복잡도가 증가할수록 제안된 모델이 선행연구들보다 예측성능이 우수함을 확인했다. 두 번째 실험은 이상치가 제거된 BPIC2012와 반복실행이 제거된 BPIC2012에 대해서 수행되었다. 이상치가 제거된 로그에서도 제안된 모델은 가장 높은 성능을 보였으며 반복실행이 제거된 로그에서는 가장 높은 성능의 모델과 근사한 성능을 보였다. 따라서 본 논문에서 제안하는 액티비티별 특징 정규화는 잔여시간 예측의 성능을 개선할 수 있는 전처리 기법임을 입증했다.

참고문헌(Reference)

[1] M. Dumas, W. M. P. van der Aalst, and A. H. ter Hofstede, Process-aware Information Systems: Bridging People and Software through Process Technology, John Wiley & Sons, 2005.

[2] C. A. Ellis, et al., "Beyond Workflow Mining," In Proc. of the International Conference on Business Process Management, pp. 49-64, 2006. https://doi.org/10.1007/11841760_5

[3] W. van der Aalst, et al., "Process Mining Manifesto," In Proc. of the International Conference on Business Process Management, pp. 169-194, 2011. https://doi.org/10.1007/978-3-642-28108-2_19

[4] N. Tax, et al., "Predictive Business Process Monitoring with LSTM Neural Networks," In Proc. of the International Conference on Advanced Information Systems Engineering, pp. 477-492, 2017. https://doi.org/10.1007/978-3-319-59536-8_30

[5] N. Navarin, et al., "LSTM Networks for Data-aware Remaining Time Prediction of Business Process Instances," In Proc. of the 2017 IEEE Symposium Series on Computational Intelligence, pp. 1-7, 2017. https://doi.org/10.1109/SSCI.2017.8285184

[6] S. Zhang, et al., "Data-based Line Trip Fault Prediction in Power Systems using LSTM Networks and SVM," IEEE Access, Vol. 6, pp. 7675-7686, 2017. https://doi.org/10.1109/ACCESS.2017.2785763

[7] W. M. P. van der Aalst, H. Schonenberg and M. Song, "Time Prediction based on Process Mining," Information Systems, Vol 32, No.2, pp. 450-475, 2011. https://doi.org/10.1016/j.is.2010.09.001

[8] M. Polato, et al., "Time and Activity Sequence Prediction of Business Process Instances," Computing, Vol. 100, No. 9, pp. 1005-1031, 2018. https://doi.org/10.1007/s00607-018-0593-x

[9] A. Rogge-Solti and M. Weske, "Prediction of Business Process Durations using Non-Markovian Stochastic Petri Nets," Information Systems, Vol. 54, pp. 1-14, 2015. https://doi.org/10.1016/j.is.2015.04.004

[10] I. Verenich, et al., "Predicting Process Performance: A White box Approach based on Process Models," Journal of Software: Evolution and Process, Vol. 31, No. 6, e2170, 2019.

[11] Business Process Model and Notation(BPMN) version 2.0, formal/2011-01-03, Object Management Group, 2011. http://www.omg.org/spec/BPMN/2.0

[12] I. Verenich, et al., "Survey and Cross-benchmark Comparison of Remaining Time Prediction Methods in Business Process Monitoring," ACM Transactions on Intelligent Systems and Technology, Vol. 10, No. 4, pp. 1-34, 2019. https://doi.org/10.1145/3331449

[13] K. P. Kim, "Functional Integration with Process Mining and Process Analyzing for Structural and Behavioral

Properness Validation of Discovered Processes from Event Log Datasets," Applied Sciences, Vol. 10, No. 4, pp. 1493, 2020.

[14] M. -J. Park and K. -H. Kim, "Control-Path Oriented Workflow Intelligence Analysis and Mining System," In Proc. of the International Conference on Convergence Information Technology, pp. 951-960, 2007. https://doi.org/10.1109/ICCIT.2007.212

[15] K. Kim, M. Yeon, B. Jeong and K. Kim, "A Conceptual Approach for Discovering Proportions of Disjunctive Routing Patterns in a Business Process Model," KSII Transactions on Internet and Information Systems, Vol. 11, No. 2, pp. 1148-1161, 2017. https://doi.org/10.3837/tiis.2017.02.030

[16] K. P. Kim, "An XPDL-Based Workflow Control-Structure and Data-Sequence Analyzer", KSII Transactions on Internet and Information Systems, Vol. 13, No. 3, pp. 1702-1721, 2019. https://doi.org/10.3837/tiis.2019.03.034

[17] H. Ahn, and K. P. Kim, "Formal Approach for Discovering Work Transference Networks from Workflow Logs," Information Sciences, Vol. 515, pp. 1-25, 2020. https://doi.org/10.1016/j.ins.2019.11.036

[18] D. -L. Pham, H. Ahn and K. P. Kim, "Discovering Redo-Activities and Performers' Involvements from XES-Formatted Workflow Process Enactment Event Logs," KSII Transactions on Internet and Information Systems, Vol. 13, No. 8, pp. 4108-4122, 2019. https://doi.org/10.3837/tiis.2019.08.016

[19] H. Ahn, D. -L. Pham and K. P. Kim, "An Experimental Analytics on Discovering Work Transference Networks from Workflow Enactment Event Logs," Applied Sciences, Vol. 9, No. 11, pp. 2368, 2019. https://doi.org/10.3390/app9112368

[20] K. Kim, et al., "An Experimental Mining and Analytics for Discovering Proportional Process Patterns from Workflow Enactment Event Logs" Wireless Networks, online published, 2018. https://doi.org/10.1007/s11276-018-01899-z

[21] J. Kim, et al., "An Estimated Closeness Centrality Ranking Algorithm and Its Performance Analysis in Large-Scale Workflow-supported Social Networks" KSII Transactions on Internet and Information Systems, Vol. 10, No. 3, pp. 1454-1466, 2016. https://doi.org/10.3837/tiis.2016.03.031

[22] M. -J. Kim, H. Ahn and M. -J. Park, "A Theoretical Framework for Closeness Centralization Measurements in a Workflow-Supported Organization," KSII Transactions on Internet and Information Systems, Vol. 9, No. 9, pp. 1454-1466, 2015. http://dx.doi.org/10.3837/tiis.2015.09.018

[23] F. Gers, A. J. Schmidhuber and F. Cummins, "Learning to Forget: Continual Prediction with LSTM," In Proc. of the 9th International Conference on Artificial Neural Networks, pp. 850-855, 1999. https://doi.org/10.1049/cp:19991218

[24] 4TU.Centre for Research Data, https://data.4tu.nl/.

[25] C. W. Günther and E. Verbeek, XES Standard Definition, Fluxicon Process Laboratories, 2014.

◐ 저 자 소 개 ◑

함 성 훈 (Seonghun Ham)
2019년 경기대학교 컴퓨터과학과 (이학사)
2019년~현재 경기대학교 컴퓨터과학과 석사과정
관심분야: 예측적 프로세스 모니터링, 잔여시간 예측, 기계학습, 프로세스 마이닝
E-mail: seonghunham@kgu.ac.kr

안 현 (Hyun Ahn)
2011년 경기대학교 컴퓨터과학과 (이학사)
2013년 경기대학교 컴퓨터과학과 (이학석사)
2017년 경기대학교 컴퓨터과학과 (이학박사)
2018년~현재 경기대학교 컴퓨터공학부 조교수
관심분야: 비즈니스 프로세스 인텔리전스, 프로세스 마이닝, 기계학습
E-mail: hahn@kgu.ac.kr

김 광 훈 (Kwanghoon Pio Kim)
1984년 경기대학교 컴퓨터과학과 (이학사)
1986년 중앙대학교 컴퓨터과학과 (이학석사)
1994년 M.S. in Computer Science, University of Colorado at Boulder
1998년 Ph.D. in Computer Science, University of Colorado at Boulder
1986년~1991년 한국전자통신연구원 TDX개발단
2007년~2016년 콘텐츠융합소프트웨어연구센터 센터장
1998년~현재 경기대학교 컴퓨터공학부 교수
1998년~현재 데이터·프로세스 공학 연구실 지도교수
2017년~현재 콘텐츠융합소프트웨어연구소 소장
2020년~현재 범죄예방능동빅데이터연구소 소장
관심분야: CSCW, 워크플로우 시스템, 비즈니스 프로세스 관리, 프로세스 마이닝, 엔터프라이즈 소셜 마이닝·분석, 프로세스 예측, 예측적 프로세스 모니터링·모델링, 능동콘텐츠 빅데이터공학기술
E-mail: kwang@kgu.ac.kr