http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.26, No.1, February 2021, pp.127-140
https://doi.org/10.7838/jsebs.2021.26.1.127

확장된 LSTM 오토인코더 기반 이상 시퀀스 탐지 기법

An Anomalous Sequence Detection Method Based on An Extended LSTM Autoencoder

이주연(Jooyeon Lee)*, 이기용(Ki Yong Lee)**

초 록

최근 센서 측정 데이터, 구매이력 등과 같이 시간 정보를 포함하는 시퀀스(sequence) 데이터가 다양한 응용에서 발생되고 있다. 주어진 시퀀스들 중 다른 시퀀스들과 매우 상이한 이상(anomalous) 시퀀스를 탐지하는 기법들은 지금까지 많이 연구되어왔으나 이들 대부분은 주로 시퀀스 내 원소들의 순서만을 고려하여 이상 시퀀스를 찾는다는 한계가 있다. 따라서 본 논문에서는 원소들의 순서와 원소들 간의 시간 간격 모두를 고려하는 새로운 이상 시퀀스 탐지 기법을 제안한다. 본 논문에서 제안하는 방법은 확장된 LSTM 오토인코더 모델을 사용한다. 이 모델은 시퀀스를 해당 시퀀스 내 원소들의 순서와 시간 간격 모두를 효과적으로 학습할 수 있는 형태로 변환하는 층을 추가로 가진다. 제안방법은 확장된 LSTM 오토인코더 모델로 주어진 시퀀스들의 특징을 학습한 뒤, 해당 모델이 잘 복원하지 못하는 시퀀스를 이상 시퀀스로 탐지한다. 본 논문에서는 정상 시퀀스와 이상 시퀀스를 혼합한 가상 데이터를 사용하여 제안 방법이 전통적인 LSTM 오토인코더만을 사용하는 방법과 비교하여 100%에 가까운 정확도를 나타냄을 보인다.

ABSTRACT

Recently, sequence data containing time information, such as sensor measurement data and purchase history, has been generated in various applications. So far, many methods for finding sequences that are significantly different from other sequences among given sequences have been proposed. However, most of them have a limitation that they consider only the order of elements in the sequences. Therefore, in this paper, we propose a new anomalous sequence detection method that considers both the order of elements and the time interval between elements. The proposed method uses an extended LSTM autoencoder model, which has an additional layer that converts a sequence into a form that can help effectively learn both the order of elements and the time interval between elements. The

이 논문은 2020년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2018R1D1A1B07045643).

* First Author, Graduate Student, Division of Computer Science, Sookmyung Women's University (onew@sookmyung.ac.kr)

** Corresponding Author, Professor, Division of Computer Science, Sookmyung Women's University (kiyonglee@sookmyung.ac.kr)

Received: 2021-01-04, Review completed: 2021-02-08, Accepted: 2021-02-19

proposed method learns the features of the given sequences with the extended LSTM autoencoder model, and then detects sequences that the model does not reconstruct well as anomalous sequences. Using experiments on synthetic data that contains both normal and anomalous sequences, we show that the proposed method achieves an accuracy close to 100% compared to the method that uses only the traditional LSTM autoencoder.

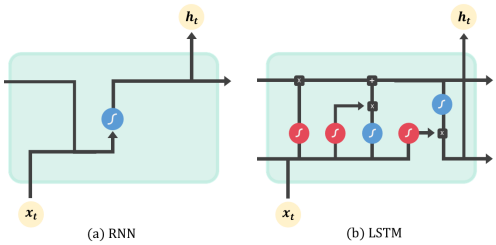
키워드 : 시퀀스, 이상치 탐색, LSTM 오토인코더
Sequence, Anomaly Detection, LSTM Autoencoder

1. 서 론

최근 온라인 서비스 및 IoT 기술 보급이 증가함에 따라 다양한 응용에서 이산적이고 상징적인 원소로 표현되는 타임스탬프(timestamp) 데이터가 활발하게 생성되고 있다. 타임스탬프 데이터는 관측된 원소의 시간 정보를 포함하는 데이터로, 이들은 그의 발생 순서에 따라 나열되어 일종의 시퀀스(sequence)를 구성한다. 센서 기반의 조명이나 온도 조절기에서 발생한 제어 신호 및 웹상에서 사용자들의 행동을 기록한 로그 파일 등과 같은 데이터는 원소들이 발생시간에 따라 나열된 시퀀스의 대표적인 예이다. 이상(anomalous) 시퀀스 탐지란 주어진 시퀀스들 중에서 대다수의 시퀀스들과 유달리 큰 편차를 나타내는 희귀한 시퀀스를 찾아내는 것을 의미한다. 이는 장치의 작동이나 사용자의 움직임이 정상적으로 수행되고 있지 않음을 나타낼 수 있으며 이를 통해 시스템 고장, 침입자 탐지, 특이현상 발견, 부정행위 적발 등에 널리 사용되고 있다[9].

최근에는 이상치 탐지(anomaly detection) 분야에 있어 딥러닝을 사용하는 기법들이 매우 활발하게 연구되고 있다[4]. 대부분의 딥러닝 기반 이상치 탐지 연구들은 주로 오토인코더(autoencoder)를 사용한다[5]. 오토인코더는 인공신경망의 한 종류로, 벡터, 다차원 배열, 시계열, 그래프, 시퀀스 등 다양한 형태의 데이터를 입력으로 가질 수 있다. 이들은 입력받는 데이터에 따라 그 내부를 구성하는 인공신경망이 달라질 수 있는데, 이상 시퀀스 탐지는 그 중에서도 LSTM(long short-term memory) 오토인코더를 사용한다[13, 14]. LSTM은 순환신경망(RNN, recurrent neural network)의 한 종류로서, 과거에 발생한 이벤트가 이후 이벤트 발생에 영향을 주는 구조로 이루어져 있다. 따라서 이들은 선후관계가 존재하는 시퀀스 데이터를 다루기에 매우 적합하다. 또한, 기존의 순환신경망은 데이터의 크기가 커질수록 비교적 먼 과거에 입력된 이벤트에 대한 학습 결과가 현재 이벤트 예측에 제대로 반영되지 않는 장기 의존성(long- term dependency) 문제를 가지고 있었다. LSTM은 이를 기존 순환신경망 셀(cell)에 구조적인 변화를 주어 해결하였다[8]. <Figure 1>의 (a)와 (b)는 기존 순환신경망과 LSTM의 셀 아키텍처로, 그들의 구조적인 차이점을 보여준다.

시퀀스의 원소들이 시간적 흐름에 따라 구성되는 경우, 이들의 발생 순서뿐만 아니라 시간 간격 역시 이상치를 판단하는데 중요한 특징이 된다. <Table 1>은 온라인 쇼핑몰에서 사용자가 발생시킨 이벤트 로그를 간략히 나타낸 예시이다. 이때, 각 이벤트의 발생 순서만을 고려한다면, 로그인, 구매, 결제, 로그아웃의 일반적인 이벤트 순서를 갖는 S1~S4는 정상 시퀀스이며



〈Figure 1〉 Difference between the Architectures of RNN and LSTM

〈Table 1〉 Example of User Event Logs

S1	login, shopping, payment, logout
S2	login, shopping, payment, shopping, payment, logout
S3	login, shopping, payment, shopping, logout
S4	login, shopping, shopping, payment, logout
S5	login, login, login, login, login, login

로그인 이벤트만이 반복적으로 일어난 S5는 이상 시퀀스이다. 그러나 원소들의 시간 간격까지 고려한다면 S1~S4 역시 이상 시퀀스의 가능성을 지니게 된다. 예를 들어, S4=<login, shopping, shopping, payment, logout>에서 각 이벤트 사이의 시간 간격이 매우 작다면 이는 이상 시퀀스로 판단되며 이를 통해 매크로를 사용한 부정구매 등을 적발할 수 있다. 하지만 이러한 시간 간격의 중요성에도 불구하고 지금까지 대부분의 이상 시퀀스 탐지 연구는 원소들의 순서만을 고려하였으며 그들 사이의 시간 간격까지 고려한 연구는 거의 이루어지지 않았다.

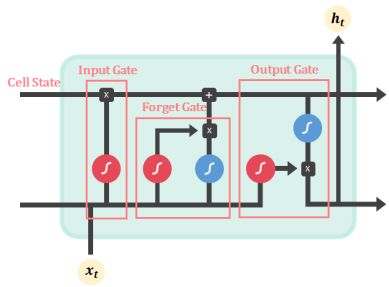
따라서 본 논문에서는 시퀀스 내 원소들의 발생 순서뿐만 아니라 원소들 사이의 시간 간격까지 고려하는 새로운 딥러닝 기반 이상 시퀀스 탐지 기법을 제안한다. 이를 위해 본 논문의 제안 방법은 기존 LSTM 오토인코더에 새로운 층(layer)을 추가하여 모델을 확장한다. 해당 층은 입력받은 데이터를 가중치 그래프로 변환하여 이를 순서대로 LSTM 층에 전달하는 역할을 수행한다. 가상 데이터를 기반으로 한 다양한 실험을 통해 제안 방법이 일반적인 LSTM 오토인코더만을 사용하는 방법보다 효과적으로 이상 시퀀스를 탐지함을 확인하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문에서 제안하는 모델을 이해하기 위해 필요한 사전 지식을 간략히 설명하고 관련 연구를 살펴본다. 제 3장에서는 LSTM 오토인코더 확장을 통한 새로운 이상 시퀀스 탐지 기법을 제안하고 이를 상세히 설명한다. 제 4장에서는 가상 데이터를 기반으로 진행한 실험 결과를 보이고, 제 5장에서 결론을 맺는다.

2. 사전 지식 및 관련 연구

2.1 LSTM

본 논문에서는 이상 시퀀스 탐지를 효과적으로 수행하기 위해 LSTM 오토인코더를 사용하며 해당 모델은 LSTM으로 LSTM 오토인코더의 각 층을 구성한다. <Figure 2>는 LSTM의 셀의 일반적인 아키텍처를 보여준다.



〈Figure 2〉 General Architecture of an LSTM Cell

LSTM은 기존 순환신경망이 가진 장기의존성 문제를 해결하기 위해 각 셀에 셀 스테이트(cell state)를 추가한다. 또한, 기존 히든 레이어(hidden layer)의 구조를 인풋 게이트(input gate), 포겟 게이트(forget gate), 아웃풋 게이트(output gate)로 대체한다. 고정 길이 T의 시퀀스 $S=<e_1, \cdots, e_t, \cdots, e_T>$가 주어졌을 때, 시간 t에서의 인풋 게이트, 포겟 게이트, 아웃풋 게이트는 다음와 같은 식 (1)~식 (3)으로 계산된다[11]. 여기서, e_t는 시간 t에서의 입력이며 W는 신경망의 가중치, b는 바이어스 항(bias term)을 나타낸다. σ는 활성화 함수를 의미한다. 모델은 현재 시점에서의 입력과 이전 히든 스테이트(hidden state) h_{t-1}로부터 전달된 시퀀스의 정보를 포함하는 히든 스테이트 벡터 h_t를 갖는다.

$$i_t = \sigma(W_{e_i}^T \times e_t + W_{h_i}^T \times h_{t-1} + b_i) \quad (1)$$

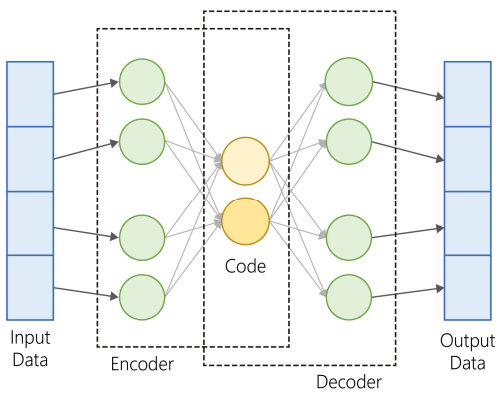
$$f_t = \sigma(W_{e_f}^T \times e_t + W_{h_i}^T \times h_{t-1} + b_f) \quad (2)$$

$$f_t = \sigma(W_{e_o}^T \times e_t + W_{ho}^T \times h_{t-1} + b_o) \quad (3)$$

포겟 게이트는 이전 단계로부터 넘어오는 과거 데이터의 영향력을 얼마나 반영할지 결정하며, 인풋 게이트는 현재 시점에서 입력된 데이터의 영향력을 결정한 다음 이 정보를 셀 스테이트에 업데이트한다. 즉, 셀 스테이트는 현재까지 입력된 모든 데이터에 대한 정보를 저장하고 있는 메모리이다. 아웃풋 게이트는 출력될 데이터를 결정하는 역할을 한다. 이러한 구조적인 변화를 통해 LSTM은 기존 순환신경망의 장기의존성 문제를 해결하였다.

2.2 오토인코더

<Figure 3>은 오토인코더의 일반적인 아키텍처를 나타낸다. 오토인코더는 인코더(encoder)와 디코더(decoder)라는 두 가지의 구조로 구성되며 입력받은 데이터의 특징을 비지도(unsupervised) 방식으로 학습한다. 이를 위해 오토인코더는 인코더를 통해 입력받은 각 데이터를 그의 특징을 잘 표현하는 저차원의 코드(code)로 변환한다. 변환된 코드는 이후 디코더를 통해 다시 원 데이터로 복원된다. 오토인코더의 상세한 학습 과정은 다음과 같다. 가장 먼저, 입력 데이터와 히든 레이어의 가중치를 계산하여 활성화 함수 통과시킨다. 본 논문에서는 오토인코더의 활성화 함수로서 시그모이드(sigmoid) 함수를 사용하였다. 활성화 함수를 통과시켜 나온 결과값은 출력층(output layer)의 가중치와 함께 계산되어 다시 한번 활성화 함수를 통과한다. 이 결과값을 사용하여 손실함수 값을 계산한다. 본 논문의 제안 모델은 손실함수로서 각 오차를 제곱한 값들의 합을 사용하였다. 다음으로 손실함수 값을 최적화하기 위해 SGD(stochastic gradient descent)을 사용하며 최종적으로 오류역전파(backpropagation)를 통해 오토인코더의 가중치를 갱신한다.



〈Figure 3〉 General Architecture of Auto-encoders

오토인코더는 원 데이터와 복원된 데이터 사이의 차이를 최소화하는 것을 목표로 원 데이터에 내재되어 있는 패턴들을 학습한다. 오토인코더를 기반으로 한 이상치 탐지 기법들은 이러한 오토인코더의 학습 목표를 역으로 이용한다. 즉, 오토인코더는 극히 적은 빈도로 발생하는 이상 데이터의 특징을 제대로 학습하지 못할 것이므로 이상 데이터의 원 데이터를 제대로 복원할 수 없게 된다. 따라서 오토인코더가 복원한 데이터와 원 데이터 간의 차이가 매우 크거나 일정 임계값 이상인 데이터를 출력하면 이상치를 매우 효과적으로 탐지할 수 있다.

본 논문에서는 이상 시퀀스 탐지를 위해 다양한 오토인코더 모델 중에서도 시퀀스를 다루기에 가장 적합한 LSTM 오토인코더를 사용한다. LSTM 오토인코더의 인코더는 입력받은 시퀀스로부터 특징을 추출하고 이를 저차원의 고정 길이 벡터로 변환하여 패턴을 학습한다. 디코더에서는 학습된 패턴을 기반으로 현재 시점의 히든 스테이트와 과거의 데이터로부터 예측된 값을 사용하여 원 시퀀스를 재구성한다.

2.3 이상 시퀀스 탐지 연구

이상 시퀀스를 탐지하는 연구는 오랜 시간에 거쳐 관심있게 다루어진 주제이며, 다방면으로 연구가 진행되었다. 본 장에서는 이성 시퀀스 탐지 연구를 크게 딥러닝을 사용한 방법과 그렇지 않은 방법으로 나누어 살펴본다.

2.3.1 딥러닝을 사용하지 않은 연구

Cao et al.[3]은 시간의 흐름에 따라 관측된 모든 데이터를 하나의 시퀀스로 인식하며 이를 대상으로 새로운 패턴 시맨틱(semantic)을 정의한다. 이에 따라 빈발하지 않은 이상 서브시퀀스(subsequence)를 찾아낸다. Wang et al.[15]은 이상 시퀀스를 탐지하기 위해 여러 개의 가지치기(pruning) 기법들을 통합하고 시퀀스의 확률론적 강도를 계산하는 휴리스틱 방식을 사용한다. Cai et al.[2]은 가중치 데이터 스트림에서 이상 시퀀스를 탐지하는 연구로, 패턴 마이닝 단계에서 최대 가중치와 최대 확률의 개념을 도입하여 탐색할 패턴의 규모를 줄인다. Boniol and Palpanas[1]은 기다란 시퀀스 내에서 이상 서브시퀀스를 탐지하기 위해 Series2Graph라는 새로운 시퀀스 표현 제안한다. 이 방법은 시퀀스를 저차원으로 임베딩(embedding)하여 하나의 그래프로 표현한다.

대부분의 딥러닝을 사용하지 않은 이상 시퀀스 탐지 연구들은 대게 하나의 커다란 시퀀스를 대상으로 하여, 이를 다양한 길이의 서브시퀀스로 분할한다. 그런 다음, 각 서브시퀀스의 빈도수를 측정하고 빈발하지 않은 서브시퀀스 집합을 생성한다. 해당 집합에 포함되는 시퀀스를 하나 이상 포함하고 있는 시퀀스가 이상 시퀀스로 판단된다. 이 연구들은 단순히 패턴의 빈도수만을 기준으로 하며 하나의 특징을 가진 시퀀스를 대상으로 이상치를 탐지하기 때문에 본 논문에서 다루고자 하는, 시퀀스 내 원소들의 발생 순서와 그 시간 간격에 대한 정보를 담고 있는 데이터에 적합하지 않다.

2.3.2 딥러닝을 사용한 연구

딥러닝을 기반으로 한 이상 시퀀스 탐지 연구는 이미 다양한 응용에서 연구되었다. Kim and Cho[10]은 웹 트래픽 데이터에서 이상치를 탐지하기 위해 새로운 인공신경망인 C-LSTM을 제안한다. 이 방법은 LSTM과 CNN 및 DNN을 결합하여 데이터에 포함된 공간 및 시간 정보를

효과적으로 추출하고 모델링한다. Li et al.[12]은 LSTM과 스택(stacked) 오토인코더를 사용하여 다중 특징 시퀀스에 대한 이상치를 탐지한 연구로서, 데이터에 라벨이 없거나 이상치에 대한 경험적 지식이 존재하지 않는 경우를 대상으로 한다. Chong and Tay[6]는 비디오 내 이상 프레임(frame)을 찾아내기 위해 CNN과 스택 컨볼루션 LSTM을 통합한다. 즉, CNN을 사용하여 장면 내 물체의 움직임에 대한 공간적 정보를 전달하며 LSTM을 사용하여 물체의 시간적 정보를 전달한다. 또한, Ghrib et al.[7]는 고차원의 시계열 데이터에서 효과적인 이상 시퀀스 탐지를 위해 LSTM 오토인코더를 사용한다. 이는 LSTM 오토인코더의 인코더를 사용하여 정상 시퀀스의 패턴을 학습한 다음, 이를 분류하기 위해 SVM(support vector machine)과 결합하여 사용한다. Zhao et al.[17] 역시 시계열 데이터에서 이상 시퀀스를 탐지하기 위해 새로운 TSAD 방법을 제안한다. 이 방법은 양방향(bidirectional) LSTM 오토인코더와 가우시안(gaussian) 분류 모델을 사용한다. 위 연구들은 시간 정보를 포함하는 데이터에서 이상 시퀀스를 탐지하기 위해 LSTM을 기반으로 하며, 시퀀스 내 원소들의 순서를 고려한다는 점에서 본 논문이 제안하는 방법과 유사하다. 하지만 이들은 모두 원소들 간의 시간 간격을 아예 고려하지 않거나 모두 동일하다고 가정하는 한계를 갖는다.

3. 제안 방법

3.1 개요

본 논문에서는 주어진 시퀀스들 중에서 대다수의 시퀀스들과 큰 편차를 가진 희귀한, 이상 시퀀스를 효과적으로 탐지하는 방법을 제안한다. 여기서, 시퀀스란 시간의 흐름에 따라 관측된 이벤트 혹은 원소들이 그의 발생 순서에 따라 나열된 데이터를 일컫는다. 그 예시로는 웹 로그, 휴대전화 애플리케이션 로그, GPS 위치 정보 등이 있다. 본 논문에서 제안하는 방법은 원소들의 순서만을 주로 고려한 기존의 방법들과는 달리 원소들 사이의 시간 간격까지 고려하여 이상 시퀀스를 탐지한다.

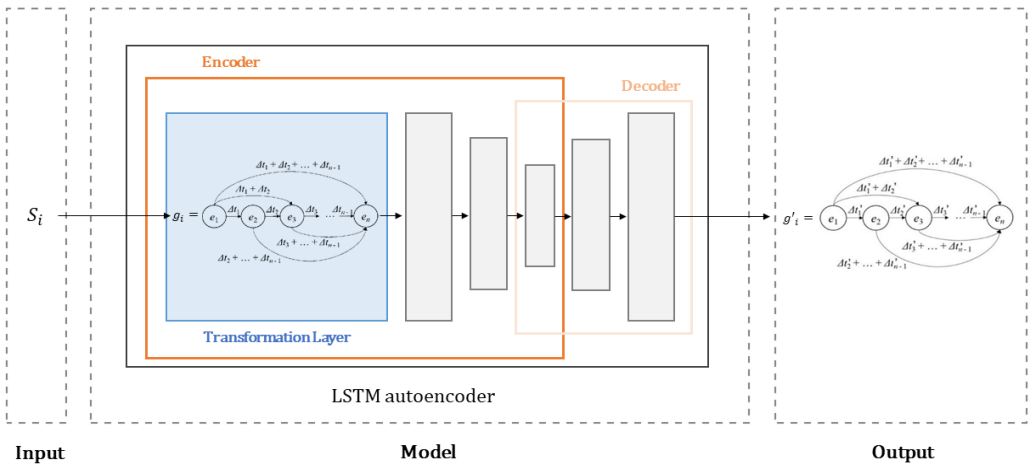
3.2 기호 및 문제 정의

N개의 시퀀스 $s_1, s_2, \cdots, s_N$이 주어졌다고 하자. 이때, 각 시퀀스 $s_i (i = 1, 2, \cdots, N)$는 원소와 해당 원소가 발생한 시간의 쌍으로 나타내어진다. 즉, 각 시퀀스는 $< (e_1, t_1), (e_2, t_2), \cdots, (e_n, t_n)>$의 형태를 가지며, 여기서 $e_1, e_2, \cdots, e_n$는 발생한 원소를, $t_1, t_2, \cdots, t_n$는 각 원소 $e_1, e_2, \cdots, e_n$가 발생한 시간을 나타낸다. 본 논문은 $s_1, s_2, \cdots, s_N$ 중 대다수의 시퀀스들과 원소들의 발생 순서 혹은 원소들 사이의 시간 간격이 상이한 이상 시퀀스를 찾는 것을 목표로 한다.

3.3 제안하는 이상 시퀀스 탐지 기법

본 절에서는 앞선 3.2절에서 정의한 기호 및 문제를 바탕으로, 시퀀스 내의 원소들의 순서뿐만 아니라 그들 사이의 시간 간격까지 고려한 효과적인 이상 시퀀스 탐지 기법을 제안한다.

본 논문의 제안 방법은 선후관계를 갖는 데이터인 시퀀스를 다루기 위해 이들에게 가장 적합한 딥러닝 모델인 LSTM 오토인코더를



〈Figure 4〉 Architecture of the Proposed Model

활용한다. <Figure 4>에서 볼 수 있듯이, 본 논문에서는 시퀀스를 입력으로 받으면 가장 먼저 선행되어, 특정한 기능을 수행하는 새로운 층을 정의한다. 이 층은 LSTM 오토인코더의 인코더 부분에 추가되며 이를 통해 모델의 확장이 이루어진다. 해당 층의 기능은 모델이 원소들의 순서 및 그 시간 간격이라는 두 가지 요소를 좀 더 효과적으로 고려하여 훈련할 수 있도록 입력받은 시퀀스의 형태를 그래프로 변환하는 것이다[16].

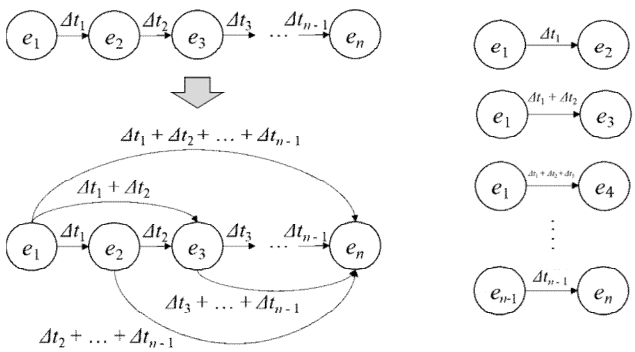
본 논문에서 제안하는 이상 시퀀스 탐지 기법은 크게 전처리, 모델 훈련 그리고 이상 시퀀스 출력의 세 단계의 과정을 거친다. 다음은 이 세 단계를 각각 자세히 설명한다.

3.3.1 전처리

전처리 단계에서는 각 시퀀스 $s_i (i = 1, 2, \cdots, N)$에 대해 그 형태를 제안 모델의 입력으로 사용할 수 있도록 변환한다. 기존의 각 시퀀스는 $s_i = <(e_1, t_1), (e_2, t_2), \cdots, (e_n, t_n)>$와 같이 원소와 해당 원소가 발생한 절대 시간의 쌍들로 이루어져 있다. 이때, 이들이 가진 절대 시간 값을 사용하여 이들이 각각을 그의 바로 앞 원소와의 시간 간격을 나타내도록 변환한다. 결과적으로 각 시퀀스는 $s_i = <(e_1, 0), (e_2, \Delta t_1), \cdots, (e_n, \Delta t_{n-1})>$의 형태를 가지게 되며, 여기서 0, Δt_1, $\cdots$, Δt_{n-1}는 각각 e_1, e_2, $\cdots$, e_n와 바로 그 앞 원소와의 시간 간격을 의미한다. 즉, $\Delta t_0 = 0$, $\Delta t_1 = t_2 - t_1$, $\Delta t_2 = t_3 - t_2$, $\cdots$, $\Delta t_{n-1} = t_n - t_{n-1}$이다. 이렇게 변환된 각 시퀀스를 <Figure 4>의 LSTM 오토인코더 모델의 입력으로 사용한다.

3.3.2 모델 훈련

앞 절에서 설명한 바와 같이 각 시퀀스 s_i가 전처리 과정을 거쳐 원소와 시간 간격의 순서쌍으로 재구성되면 이를 <Figure 4> 모델의 입력으로 사용한다. 본 논문에서 제안하는 모델의 기반이 되는 LSTM 오토인코더는 시퀀스를 다루기에 최적화된 딥러닝 모델로, 본 논문에서는 원소들의 순서 및 그들 사이의 시간 간격을 보다 효과적으로 훈련하기 위해 이를 확장하여 사용한다.



〈Figure 5〉 Generation of Graph g_i from Sequence s_i

모델의 확장은 훈련을 위한 LSTM층으로만 구성된 기존의 인코더에 새로운 기능을 수행하는 층을 추가하는 것으로 이루어지며, 본 논문에서는 이를 변환층(transformation layer)라 정의한다. 변환층은 입력으로 들어온 시퀀스 s_i로부터 s_i 내 원소들 사이의 시간 간격을 표현하는 그래프인 g_i를 생성한다. 그 후, 생성된 그래프를 다시 간선의 개수가 1인 부분 그래프 단위로 분할하며, 이를 LSTM 층에 입력으로 전달한다. 변환층이 시퀀스 s_i로부터 그래프인 g_i를 생성하는 구체적인 과정은 다음과 같다.

<Figure 5>의 (a)는 주어진 시퀀스 si로부터 그래프 g_i를 생성하는 방법을 보여준다. 제안 방법은 먼저 시퀀스 내에서 인접한 원소들 사이의 시간 간격을 나타내도록 변환된 시퀀스 s_i = <(e_1, 0), (e_2, Δt_1), …, (e_n, Δt_{n-1})>를 <Figure 5>(a)의 상단 그래프와 같이 표현한다. 그리고 이 그래프의 각 노드를 그의 뒤에 나오는 모든 노드들과 연결하여 <Figure 5>(b)의 하단과 같은 그래프를 생성한다. 이때 각 간선의 가중치로는 두 노드 사이에 존재하는 모든 원소들 사이의 시간 간격의 총합을 부여한다. 예를 들어, 두 노드 e_i와 e_j(단, $i < j$)를 연결하는 간선의 가중치는 $\Delta t_1 + \Delta t_{i+1} + \cdots + \Delta t_{j-1}$가 된다. 이렇게 함으로써 그래프 gi는 시퀀스 s_i 내 임의의 원소들 간 시간 간격을 모두 표현하게 된다.

최종적으로 생성된 그래프 g_i는 <Figure 5>의 (b)와 같이 간선의 개수가 1인 부분 그래프 단위로 분할된다. 이때, 분할된 그래프들은 원 시퀀스 내 원소들의 순서대로 나열된다. 즉, 가장 먼저 발생한 원소 e_1가 존재할 때, 전체 그래프는 <(e_1, e_2, Δt_1), (e_1, e_3, $\Delta t_1 + \Delta t_2$), …, (e_{n-1}, e_n, Δt_{n-1})>의 순서를 갖는다. 이렇게 나열된 전체 그래프는 분할층의 출력으로서 나와 LSTM 층의 입력으로 들어가며, 제안 모델의 훈련 데이터로 사용된다.

본 논문에서는 <Figure 5>의 제안 모델을 훈련할 때, 손실함수로 MSE(mean squared error)를 사용한다. MSE는 아래 식 (4)와 같이 정의되며, 모든 데이터 N에 대해 모델이 예측한 값인 $\hat{y}_i$와 실제 값인 y_i사이의 평균 제곱 오차를 계산한다. 각 시퀀스에 대한 오차들을 모두 합하여 이를 총 손실로서 사용한다.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \qquad (4)$$

3.3.3 이상 시퀀스 출력

모델의 학습이 완료되면, 마지막 단계로 입력 시퀀스들 중 이상 시퀀스로 판단된 시퀀스를 출력한다. 이를 위해 본 논문의 제안 방법은 각 시퀀스 s_i에 대해 다음과 같은 작업을 수행한다. 가장 먼저, 기존 시퀀스 s_i로부터 생성된 그래프 g_i를 다시 모델에 입력하며 이를 통해 모델로부터 g_i를 복원한 결과인 g_i'를 획득한다. 모든 그래프 g_i에 대하여 원 그래프와 복원된 그래프 사이의 재구성 오차(reconstruction error)값을 계산한다. 계산된 재구성 오차 값에 대하여 이 값이 가장 크거나 일정 임계값 이상인 시퀀스를 이상 시퀀스로서 출력한다. 이 일련의 과정은 <Figure 6>과 같은 알고리즘으로 나타낼 수 있다.

본 논문에서 사용한 재구성 오차 값은 원 데이터와 복원된 데이터 사이의 평균 제곱 거리를 사용한다. 결론적으로 본 논문의 제안 방법은 제안 모델이 복원한 원소들의 순서 및 원소들 간의 시간 간격이 원 시퀀스와 크게 다른 시퀀스들을 이상 시퀀스로 판단하여 출력한다.

```
Input :
    • Graph g_i generated from original sequence S_i
    • Trained LSTM autoencoder model

Parameter: k or threshold
Begin
    Error = {}
    Anomaly = {}
    i ← 1 while i ≤ n do
        Input each graph g_i into model
        Get reconstructed graph g'_i
        Compute reconstruction error e_i between g_i and g'_i
        Error = Error+e_i
    end
    for each e_i in Error do
        if e_i > k or threshold
        Anomaly = Anomaly + S_i
    end
End

Output : sequence S_i in Anomaly
```

〈Figure 6〉 Algorithm for Printing Anomalous Sequence

4. 실험 결과

본 장에서는 본 논문에서 제안한 이상 시퀀스 탐지 기법에 대해 성능을 측정 결과를 보인다. 첫 번째로는 주어진 전체 시퀀스 내 이상 시퀀스의 비율을 조금씩 증가시켜가며 제안 방법의 성능을 측정하였다. 두 번째로는 주어진 시퀀스들의 평균 길이를 조금씩 늘려가며 제안 방법의 성능을 측정하였다. 측정한 성능을 다른 모델과 비교함으로써 본 논문이 제안하는 모델의 정확도 및 효과성을 보인다.

4.1 실험 환경 및 방법

본 논문에서 제안하는 이상 시퀀스 탐지 방법은 Python 3.6.10과 PyTorch1.5.1를 사용하여 구현되었다. 모델 훈련 및 알고리즘 실행은 NVIDIA GeForce RTX 2080TI와 Intel i9-9900K 3.60 GHz CPU, 32GB RAM, 500GB SSD, 2TB HDD가 탑재된 PC에서 수행하였다.

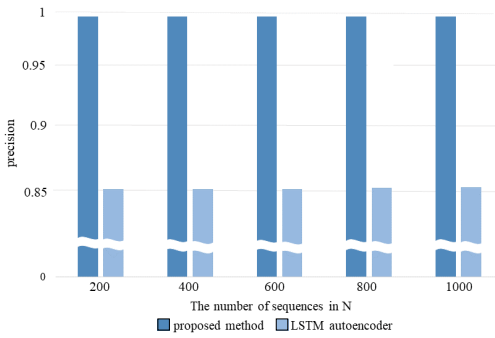
본 연구는 기존의 이상 시퀀스 탐지 연구들이 시퀀스 내 원소들의 순서만을 고려했던 것과는 달리 원소들 사이의 시간 간격도 함께 고려한다. 따라서 각 원소는 발생한 원소 그 자체와 더불어 해당 원소가 발생한 시간이라는 두 가지의 값을 가져야 한다. 이를 위해 본 논문은 가상 데이터를 생성하여 실험을 진행하였다. 가상 데이터의 구체적인 생성 방법은 다음과 같다. 우선 길이가 L인 시퀀스 S를 임의로 생성하였다. 이때, 시퀀스 S를 구성하는 각 원소는 미리 주어진 10개의 이벤트 중에서 임의로 지정하였으며, 그들 사이의 시간 간격이 1~20초 범위 안에 있도록 하였다. 그런 다음, 시퀀스 S에 대해 임의로 (이벤트, 발생시간) 쌍을 조금 추가하거나, 삭제 혹은 변경함으로써 S와 비슷한

대량의 시퀀스 집합 N을 생성하였다. 이 N은 서로 유사한, 정상 시퀀스 집합을 나타낸다. 또한, 정상 시퀀스 집합 N과 비교하여 소량의 이상 시퀀스 집합을 나타내는 A1과 A2를 각각 생성하였다. A1은 정상 시퀀스 내의 원소들과 순서 및 시간 간격이 모두 크게 다르며, A2는 원소들의 순서 자체는 유사하지만 그들의 시간 간격이 매우 상이하다. 실험에서는 A1과 A2에 속한 시퀀스의 개수를 각각 10개로 고정하고 N에 속한 시퀀스의 개수를 200, 400, 600, 800, 1000개로 증가시켜가며 A1∪A2에 속한 시퀀스를 이상 시퀀스로 탐지하는 성능을 측정하였다. 이는 전체 시퀀스 집합 내에서 이상 시퀀스가 차지하는 비율이 각각 10%, 5%, 3.3%, 2.5%, 2%로 변화함을 나타낸다. 또한, N, A1, A2에 속한 시퀀스들의 평균 길이 L을 5, 10, 15, 20, 25로 증가시켜가면서 A1∪A2에 속한 시퀀스를 이상 시퀀스로서 탐지하는 성능도 측정하였다.

성능 평가 척도로는 정밀도(precision)를 사용하였다. 정밀도는 모델이 N∪A1∪A2에 속하는 모든 시퀀스들에 대해 이상 시퀀스 탐지 기법이 이상 시퀀스라고 판단한 것 중 실제 A1∪A2에 포함된 시퀀스들의 비율을 나타낸다. 즉, 이상 시퀀스 탐지 기법이 얼마나 정확하게 이상 시퀀스를 판단하였는지를 보여준다. 본 논문에서는 LSTM 오토인코더를 확장하여 사용한 제안 방법과 단순 LSTM 오토인코더를 사용한 방법 간의 정밀도를 측정하였으며, 이를 통해 각 방법이 이상 시퀀스를 얼마나 효과적으로 찾아내는지 비교 분석하였다.

4.2 이상 시퀀스 비율 변화에 따른 성능 평가

첫 번째 실험에서는 전체 시퀀스들 내 이상 시퀀스가 차지하는 비율을 조금씩 변동시켜가면서 제안 방법의 성능을 단순 LSTM 오토인코더를 사용한 방법과 비교하였다. 이를 위해 4.1절에서 설명한 것과 같이 이상 시퀀스 집합인 A1과 A2에 속하는 시퀀스의 개수를 각각 10개로 고정하였으며, 정상 시퀀스 집합 N에 속하는 시퀀스의 개수를 200개부터 시작하여 400, 600, 800, 1000개로 늘려가며 A1∪A2에 해당하는 시퀀스를 이상 시퀀스로 판단, 탐지해 내는 성능을 측정하였다. A1과 A2에 속하는 시퀀스의 개수는 고정되어 있으므로 N의 크기가 커질수록 이상 시퀀스의 비율이 낮아짐을 알 수 있다.



〈Figure 7〉 Precision Comparison Result with Varying Ratio of Anomalous Sequences

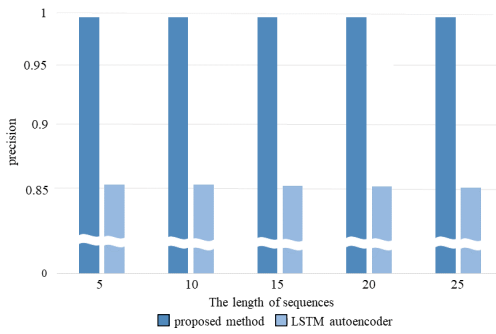
<Figure 7>은 전체 시퀀스 내에서 이상 시퀀스가 차지하는 비율을 변화시켜가며, 본 논문의 제안 방법과 단순 LSTM 오토인코더를 사용한 방법의 성능을 측정하고 이를 비교한 결과이다. 정상 시퀀스의 수가 증가할수록 모델은 정상적인 패턴을 더 많이 학습할 수 있기 때문에 두 방법 모두 정밀도가 조금씩 증가하는 경향을 보인다. 하지만 본 논문의 제안 방법은 모든 경우에 대해 단일 LSTM 오토인코더를 사용하는 방법보다 항상 우월한 성능을

보여준다. 이것은 제안 방법이 단일 LSTM 오토인코더를 사용하는 방법에 비해 이상 시퀀스 집합 A1과 A2에 속하는 이상 시퀀스들을 모두 더 잘 탐지해냄을 의미한다. 이는 제안 방법이 시퀀스 내에 존재하는 모든 원소들 사이의 시간 간격을 그래프의 구조로 표현하고 이를 분할하여 보다 정확하게 학습하기 때문이다. 따라서 제안 방법은 시퀀스 내 원소들의 시간 간격까지 고려하여 이상 시퀀스를 효과적으로 탐지함을 확인할 수 있다.

4.3 시퀀스 길이 변화에 따른 성능 평가

두 번째 실험에서는 주어진 시퀀스들의 평균 길이에 변화를 주면서 제안 방법과 단순 LSTM 오토인코더를 사용한 방법의 성능을 평가하였다. 이를 위해 정상 시퀀스 집합 N과 이상 시퀀스 집합 A1, A2에 속한 시퀀스들의 평균 길이를 5, 10, 15, 20, 25로 증가시켰으며 A1∪A2에 해당하는 시퀀스를 이상 시퀀스로서 탐지하는 성능을 측정하였다.

<Figure 8>은 시퀀스의 평균 길이를 늘려가면서 본 논문의 제안 방법과 단순 LSTM 오토인코더를 사용한 방법의 성능을 비교한 결과이다. 시퀀스들의 평균 길이가 길어질수록 원소들의 순서 및 시간 간격에 발생하는 패턴의 경우의 수가 증가하기 때문에 두 방법의 정밀도가 아주 미세하지만 조금씩 감소하는 경향을 보인다. 하지만 이 경우에도 본 논문의 제안 방법은 단일 LSTM 오토인코더를 사용하는 방법보다 항상 우수한 성능을 보여준다. 이것은 첫 번째 실험에서와 동일한 이유로, 제안 방법이 입력 시퀀스를 그래프의 구조로 변환하고 이를 훈련 데이터로 사용함으로써 원소들의 발생 순서



〈Figure 8〉 Precision Comparison Result with Varying Length of Sequences

및 시간 간격을 좀 더 정확히 학습하기 때문이다. 다시 말해, 제안 방법은 단순 LSTM 오토인코더를 사용한 방법에 비해 시퀀스 내에서 인접한 원소들 간의 시간 간격 뿐만 아니라 비교적 멀리 있는 원소들 간의 시간 간격도 정확히 학습하며 이를 통해 이상 시퀀스를 더 잘 찾아낸다.

5. 결 론

본 논문에서는 시간의 흐름이 반영된 원소들로 이루어진 시퀀스 집합에 대해, 이들 중 대다수의 시퀀스들과 유달리 큰 편차를 나타내는 희귀한 시퀀스인 이상 시퀀스를 효과적으로 탐지하는 새로운 기법을 제안하였다. 본 논문이 제안하는 방법은 기존 이상 시퀀스 탐지에 대한 대부분의 연구들이 시퀀스 내 원소들의 순서만을 고려했던 것과는 달리, 모든 원소들 사이의 시간 간격까지 함께 고려하여 이상 시퀀스를 정확히 탐지해낸다.

이를 위해 본 논문의 제안 방법은 LSTM 오토인코더 기반으로 이를 확장하여 사용한다.

모델의 확장은 기존 LSTM 오토인코더의 인코더가 보통 LSTM 층으로만 구성되던 것과는 달리, 새로운 역할을 수행하는 층을 추가함으로써 이루어진다. 본 논문에서는 이를 변환층이라 부르며, 인코더에 들어온 시퀀스는 입력층을 거쳐 가장 먼저 변환층을 통과하게 된다. 이 층은 모델이 원소의 순서와 시간 간격이라는 두 가지의 요소를 효과적으로 학습할 수 있도록 입력받은 시퀀스를 토대로 그래프를 형성한다. 또한, 형성된 그래프를 분할하여 LSTM 층에 전달함으로써 이들을 훈련 데이터로 사용할 수 있게 해준다. 모델의 학습이 완료되면 원 시퀀스를 학습된 모델에 다시 입력하여 복원된 시퀀스를 얻고 복원된 시퀀스 중 원 시퀀스와의 차이가 큰 시퀀스로 이상 시퀀스로서 출력한다.

본 논문에서는 가상 데이터를 통한 다양한 실험을 통해 본 논문에서 제안하는 방법이 단일 LSTM 오토인코더만을 사용하는 방법보다 효과적으로 이상 시퀀스를 탐지함을 확인하였다. 무엇보다 제안 방법은 입력 시퀀스 내 이상 시퀀스의 비율이나 입력 시퀀스의 길이에 상관없이 평균 0.99 이상의 매우 우수하고 일관적인 정밀도를 보여주었다. 따라서, 본 논문의 제안 방법은 시퀀스 내 원소들의 순서와 그들 사이의 시간 간격까지 고려하여, 보다 효과적이고 정확하게 이상 시퀀스를 탐지한다.

References

[1] Boniol, P. and Palpanas, T., “Series2Graph: graph-based subsequence anomaly detection for time series,” Proc. VLDB Endow, Vol. 13, No. 2, pp. 1821-1834, 2020.

[2] Cai, S., Li, L., Li, Q., Li, S., Hao, S., and Sun, R., “UWFP-Outlier: an efficient frequent-pattern-based outlier detection method for uncertain weighted data streams,” Applied Intelligence, Vol. 50, pp. 3451-3470, 2020.

[3] Cao, L., Yan, Y., Madden, S., Rundensteiner, E. A., and Gopalsamy, M., “Efficient discovery of sequence outlier patterns,” Proc. VLDB Endow, pp. 920-932, 2019.

[4] Chalapathy, R. and Chawla, S., “Deep learning for anomaly detection: A survey,” [online] Available: http://arxiv.org/abs/1901.03407, 2019.

[5] Chen, Z., Yeo, C. K., Lee, B. S., and Lau, C. T., “Autoencoder-based network anomaly detection,” ireless Telecommunications Symposium, pp. 1-5, 2018.

[6] Chong, Y. S. and Tay, Y. H., “Abnormal Event Detection in Videos Using Spatio-temporal Autoencoder,” Advances in Neural Networks-ISNN 2017, pp. 189-196, 2017.

[7] Ghrib, Z., Jaziri, R., and Romdhane, R., “Hybrid approach for Anomaly Detection in Time Series Data,” 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-7, 2020.

[8] Hochreiter, S. and Schmidhuber, J., “Long short-term memory,” Neural computation, Vol. 9, No. 8, pp. 1735-1780, 1997.

[9] Hu, J., Yang, B., Guo, C., and Jensen, C.

S., “Risk-aware path selection with time-varying, uncertain travel costs: a time series approach,” The VLDB Journal, Vol. 27, pp. 179-200, 2018.

[10] Kim, T.-Y. and S.-B. Cho., “Web traffic anomaly detection using C-LSTM neural networks,” Expert Systems with Applications, Vol. 106, pp. 66-76, 2018.

[11] Lee, D. H. and Kim, K. H., “A LSTM Based Method for Photovoltaic Power Prediction in Peak Times Without Future Meteorological Information,” The Journal of Society for e-Business Studies, Vol. 24, No. 4, pp. 119-133, 2019.

[12] Li, Z., Li, J., Wang, Y., and Wang, K., “A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment,” Int. J. Adv. Manuf. Technol., Vol. 103, pp. 499-510, 2019.

[13] Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G., “LSTM-based encoder-decoder for multi-sensor anomaly detection,” Proc. Anomaly Detection Workshop 33rd Int. Conf. Mach. Learn, 2016.

[14] Provotar, O. I., Linder, Y. M., and Veres, M. M., “Unsupervised Anomaly Detection in Time Series Using LSTM-Based Auto-encoders,” IEEE International Conference on Advanced Trends in Information Theory (ATIT), pp. 513-517, 2019.

[15] Wang, T., Duan, L., Dong, G., and Bao, Z., “Efficient Mining of Outlying Sequence Patterns for Analyzing Outlierness of Sequence Data,” ACM Transactions on Knowledge Discovery, Vol. 14, No. 5, 2020.

[16] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S., “A Comprehensive Survey on Graph Neural Networks,” IEEE Transactions on Neural Networks and Learning Systems, 2020.

[17] Zhao, J., Li, Y., He, H., and Deng, F., “One-step Predictive Encoder-Gaussian Segment Model for Time Series Anomaly Detection,” 2020 International Joint Conference on Neural Networks, pp. 1-7, 2020.

저 자 소 개

이주연 (E-mail: onew@sookmyung.ac.kr)
2019년 숙명여자대학교 소프트웨어학부 (학사)
2019년～현재 숙명여자대학교 컴퓨터과학과 (석사과정)
관심분야 데이터마이닝

이기용 (E-mail: kiyonglee@sookmyung.ac.kr)
1998년 KAIST 전산학과 (학사)
2000년 KAIST 전산학과 (석사)
2006년 KAIST 전산학과 (박사)
2006년～2008년 삼성전자 소프트웨어연구소 책임연구원
2008년～2010년 KAIST 전산학과 연구조교수
2014년～현재 숙명여자대학교 소프트웨어학부 교수
관심분야 데이터베이스, 데이터마이닝, 빅데이터, 데이터스트림