Regular paper | Journal of the KIECS. pp. 1309-1318, vol. 19, no. 6, Dec. 31. 2024, t. 128, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2024.19.6.1309

오픈소스 기반 데이터 스크래핑 탐지 시스템 설계 및 구현

이지율* · 이용주**

Design and Implementation of Data Scraping Detection System based on Open Source

Ji-Yul Lee* · Yong-Ju Lee**

요 약

데이터는 4차산업 혁명과 디지털 전환의 핵심 요소이다. 그러나 불법적으로 웹스크래핑을 통해 수집한 데이터는 소유자와 수집자간의 분쟁을 일으키고 있다. 데이터 소유자의 권리를 보호하기 위해 오픈소스를 기반으로 웹스크래핑을 탐지하는 시스템을 디자인하고 구현하였다. 선형회귀를 스크래핑 탐지 모델에 사용하였으며, 입력값으로 스크래핑을 시도하는 전후 시간 간격을 사용하였다. 반복적인 패턴은 선형회귀에서 일직선의 형태를 띄며 입력값이 많을수록 기울기는 0에 수렴하게 된다. 또한 선형회귀는 랜덤한 조회시간 간격의 범위를 쉽게 파악할 수 있다. 분석시스템은 DBMS간 실시간 데이터 동기화 기술을 사용한다. 이는 웹서버와 분석서버를 분리하여 부하를 분산시킨다. 데이터 분석 및 시각화 구현을 위해 R과 Shiny 패키지를 사용하는데, 이는 복잡한 분석 기능을 쉽게 제공하고 반응형 웹 대시보드를 제공한다.

ABSTRACT

Data is very important factor of the 4th industrial revolution and digital transformation. However, data illegally collected through Web-scraping creates disputes between data owners and scrapers. In order to protect the rights of data owners, I study methods to detect malicious Web-scraping, and design and implement open source based data scraping detection system. Linear regression is used as a scraping detection model. As an input value, time interval records of scraping attempt between before and after are used. A repetitive pattern takes the form of a straight line which is linear regression, and as the number of inputs increases, the slope converges to zero. Additionally linear regression makes it easy to infer the range of randomly query time intervals. The analysis system uses the current data capture technology which synchronizes data by transmitting data to heterogeneous DBMS in real time. This has the effect of distributing the load by separating the Web server and analysis server. R and Shiny package is used for data analysis and visualization, It supports analysis packages that implement complex functions simply and develop responsive web dashboards quickly without much knowledge of the Web development.

키워드

Data Scraping, Linear Regression, R/Shiny, Real-time Data Capture
데이터 스크래핑, 선형회귀, R/Shiny, 실시간 데이터 전송

* 경북대학교 IT대학 컴퓨터학부
(jiyul7@daum.net)
** 교신저자 : 경북대학교 IT대학 컴퓨터학부
• 접 수 일 : 2024. 09. 08
• 수정완료일 : 2024. 10. 25
• 게재확정일 : 2024. 12. 12
• Received : Sep. 08, 2024, Revised : Oct. 25, 2024, Accepted : Dec. 12, 2024
• Corresponding Author : Yong-Ju Lee
Dept. Computer Science and Engineering, Kyungpook National University
Email : yongju@knu.ac.kr

Ⅰ. 서 론

데이터는 4차 산업혁명과 디지털 전환의 핵심이자 제2의 원유라 불리고 있으며, 이러한 데이터를 얼마나 확보하는가는 국가의 경쟁력을 좌우하는 중요한 척도가 되었다. 정부는 “데이터 댐을 통한 데이터 생태계 활성화”를 목표로 디지털 산업 육성을 위한 정책을 추진하고 있다[1]. 또한, 2022년 “데이터 산업진흥 및 이용촉진에 관한 기본법”이 제정되어 데이터 경제 활성화를 제도적으로 뒷받침하고 있다[2]. 그런데 데이터 산업이 확산됨에 따라 타인의 데이터를 웹 스크래핑 기술로 무단 취득하는 행위에 의해 자산으로서의 데이터에 대한 저작권 침해 문제가 발생하여 법적 분쟁으로 이어지고 있다[3].

본 논문에서는 홈페이지를 통해 데이터를 제공하는 업체의 무형자산인 데이터를 안전하게 보호하기 위하여 홈페이지 사용자 모니터링을 통해 불법으로 데이터를 무단 취득하는 스크래핑 상황을 탐지 및 식별하는 방안을 제안한다. 이를 위해 홈페이지 사용 로그를 기반으로 스크래핑 여부를 판단하는 분석 모델을 설계 및 구현하여 모델을 검증한다.

스크래핑 탐지 분석 모델[4]이 사용하는 통계기법으로 회귀모형을 사용한다. 데이터 조회를 자료 수집 프로그램이 수행하므로 유사한 행위 패턴이 반복적으로 일어나기 때문에 흐름을 파악할 수 있는 회귀모형이 적합하다. 그리고 해당 모형을 실현하기 위해 탐지 모형을 가동하고 그 결과를 보여줄 수 있는 자동 모니터링 시스템 아키텍처를 설계한다. 홈페이지 사용 로그를 저장하는 원천 데이터베이스와 이를 분석하기 위한 타겟 데이터베이스를 두고, 원천 데이터베이스를 실시간으로 타겟 데이터베이스에 전송하고자 오픈소스인 Kafka[5]를 사용한다. 이를 통해 원천 데이터의 row 단위 변동 내용을 실시간으로 탐지 시스템으로 전송시킬 수 있다. 통계용 언어인 R[6]로 탐지 모형을 구현하고 입수한 데이터로 스크래핑 여부를 탐지한다. 분석 결과는 R의 Shiny 패키지[7]를 사용하여 웹 대시보드 시각화를 제공한다.

본 논문의 구성은 2장에서 특허로 제안된 기존 사례를 분석하고, 3장에서 스크래핑 탐지 시스템을 설계하고 4장에서 시스템을 구현하고 분석결과를 시각화한다. 그리고 5장에서 결론을 내린다.

Ⅱ. 관련 연구

스크래핑 여부를 탐지하는 기술적 방법에 대한 국내 특허로 Fingerprint 방식이 있다[8]. 웹 사용자에 대한 식별값은 브라우저의 종류, 버전, 언어, 운영체제, 쿠키사용 여부, 화면 해상도, 보유 글꼴 및 색상 등으로, 브라우저를 통해 사용자 단말 정보를 추출할 수 있는 속성값이다. 데이터를 제공하는 웹서버의 웹페이지 내에 공통 도메인에 속하는 스크래핑 검증 스크립트 파일의 URL을 삽입하고, 사용자가 접속할 때마다 브라우저에서 검증 스크립트가 실행되도록 한다. 전체 시스템 구성도는 그림 1과 같다.

전체 흐름은 ① 각 데이터 제공자의 웹 페이지에 스크래핑 검증 스크립트를 사전에 배포하여 삽입시켜 놓는다. 이 스크립트는 해당 웹페이지의 도메인에 속하거나 약속한 공통 도메인에 속하는 스크립트로 <script src=“https://test.com/validation.js“></script>와 같은 형태로 웹 페이지에 삽입할 수 있다. ② 사용자가 웹 페이지에 접속할 때마다, ③ 공통 도메인으로부터 검증 스크립트를 로딩하여 실행된다. 그리고 ④ 실행될 때마다 브라우저 내 공통 도메인 웹 저장소에 스크립트 실행과 관련된 로그를 저장한다. 이렇게 하여 사용자가 접속한 모든 웹페이지에 대한 접근기록을 남기게 한다. 한편 ⑤ 데이터 분석 서버는 검증 스크립트의 실행 관련 로그를 수신하여 사용자가 웹 스크래퍼인지 여부를 판단한다.

사용자가 웹 브라우저로 검증 스크립트가 삽입되어 있는 웹 페이지에 접속하면 스크립트가 실행된다. 이때 스크립트는 최초 실행된 시각, 설정된 시각 이후 상기 스크립트가 실행된 횟수, 최초 실행된 시각 이후 실행된 총 횟수, 그리고 마지막 단위 시간 값을 웹 저장소에 저장한다. 저장된 값들은 공격자에 의해 삭제는 가능하나 복제나 변조는 불가능하다.

데이터 분석 서버는 스크립트가 전송한 로그를 수신받고 스크래핑 여부를 분석한다. 서버는 기설정한 단위 시간 동안 상기 검증 스크립트가 실행된 횟수가 제1 임계치를 초과하는 경우 해당 사용자를 스크래퍼로 판단할 수 있다. 그러나 스크래퍼는 탐지를 회피하기 위해 단위 시간 동안 제1 임계치를 초과하지 않는 범위에서 웹 스크래핑을 계속 반복 실행할 수 있다. 그러므로 단위 시간 동안 실행된 횟수만으로 스크래

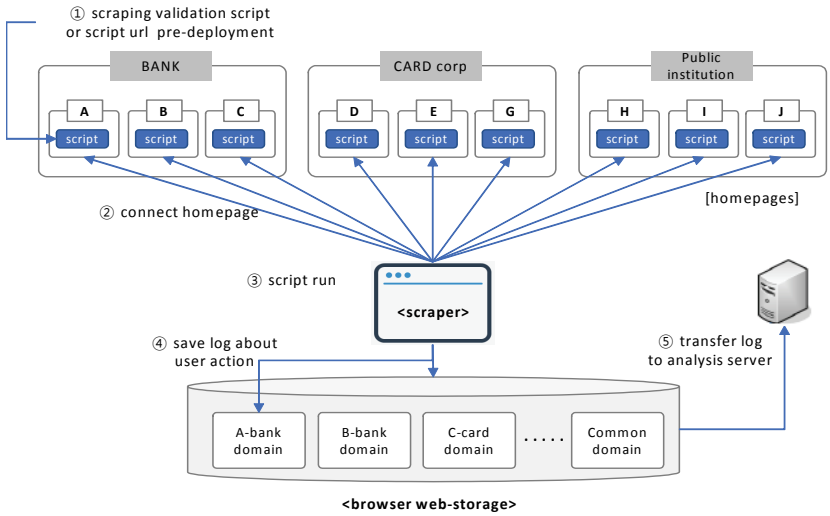

그림 1. Fingerprint 방식 스크래핑 탐지 시스템 구성도

Fig. 1 System Configuration of fingerprint method scraping detection

핑 판단 여부를 내리지 않고, 제2 임계치인 총 실행 횟수를 고려하면 스크래핑 여부를 좀 더 높은 확률로 탐지할 수 있다.

Fingerprint 방식은 단위 시간에 실행된 스크립트 횟수를 기준으로 스크래핑 여부를 판단하고 있는데, 직전 실행 시점과 현재 실행 시점 간 시간 간격 정보를 사용하지 않아 스크래핑 패턴에 대한 정보를 파악하기 어렵다. 즉, 2단계 임계치를 총 실행 횟수로 잡고 스크래핑 판단요소로 사용하고 있는데, 이는 집중적으로 실행된 스크래핑 활동인지 정상적인 사용 횟수가 많았는지에 대한 판단이 어려운 면이 있다.

III. 스크래핑 탐지 시스템 설계

3.1 스크래핑 탐지 모델

본 연구에서 제안하는 스크래핑 탐지 모델은 식(1)과 같이 선형회귀(linear regression)를 적용하였다. 선형회귀는 인과관계를 가지고 있는 두 변량 간의 함수관계를 통계적으로 규명하는 분석 방법이다. 스크래핑 탐지 모델에서 http request 순번을 제1 변량(x값)으로 사용하고, 다음 request와의 시간 차이를 제2 변량(y값)으로 하여 두 변수를 사용한다. 일반적인 스크래핑 패턴으로 요청이 들어오게 되면 시간 차이를 나타내는 y축의 값은 상수로 일정하거나 상/하단 범위를 갖는 랜덤값이므로, 선형회귀의 직선 기울기(회귀계수)는 점차적인 증가나 감소의 형태를 띄지 않고 0에 수렴하여 x축에 평행하게 될 것이다. 변량 데이터가 많이 확보되어 기울기가 0으로 판단할 정도가 되면 이는 사용자의 패턴이 반복되고 있다고 볼 수 있으며, y절편값은 반복주기의 평균을 의미한다. 일반적으로 회귀분석은 종속변수의 예상값을 목표로 하나 스크래핑 탐지 모형에서는 기울기(회귀계수)가 0에 수렴하는지를 알아내는 것이 핵심이다.

$$\text{가설함수 } y = ax + b \quad \cdots (1)$$

여기서, a는 회귀계수(regression coefficient), x는 제1 변량 독립변수(request sequence), y는 제2 변량 종속변수(time interval), 그리고 b는 y절편(intercept)이다.

회귀분석에서 최적의 a와 b를 구하기 위해 실제값과 예상값의 차이(오차)가 최소가 되도록 식(2)와 같이 손실값 cost(a,b)을 구해야 한다. 오차의 잔차 제곱합(RSS: Residual Sum of Squares)을 손실함수라 하며, 보통 잔차 제곱 합의 평균을 낸 평균 제곱 오차(MSE: Mean Square Error)로 손실을 측정한다.

$$Cost = \frac{1}{N}\sum_{i=1}^{n}(\overline{y_i} - y_i)^2 = \frac{1}{N}\sum_{i=1}^{n}(ax_i + b - y_i)^2 \quad \cdots (2)$$

여기서, N은 평균을 위한 개수를 의미하며, $(ax_i + b)$는 가설값을 의미한다. 이 가설값에서 실측

값(y_i)을 뺀 값이 오류(손실)값이다. 이를 제곱 처리하여 차이가 클수록 페널티를 적용하고 데이터셋 개수(N)로 나누어 평균 오차값을 구한다. 이제 해야 하는 일은 기울기(a)를 변화시켜 오차율이 어떻게 변하는지를 알고 이 오차를 최소화하는 방법을 찾아야 하는데, 위 손실함수에서 $\sum_{i=1}^{n}(ax_i+b-y_i)^2$의 최소값을 찾는 것이 목표이다. 선형회귀 분석에서 최소 손실을 구하는데 사용하는 방법은 종속변수 y의 평균값을 y절편으로 하고, x축에 평행한(기울기 0) 직선에서부터 시작해서 조금씩 기울기를 조절하며 회전을 시켜 RSS가 가장 작은 포인트를 찾으면 된다. 회전을 하다 보면 RSS가 다시 커지기 시작하는 포인트가 있으므로 최소 손실 포인트를 알 수 있다.

그러나 보통은 평균 제곱 오차를 이용하여 최소 손실을 구하는 데는 머신러닝 학습에 사용되는 경사 하강법(gradient descent)을 이용한다[9]. 평균 제곱 오차가 2차 함수의 곡선 형태이기 때문에 최소값에 갈수록 접선의 기울기는 0에 가까워지게 된다. 경사 하강법은 접선 기울기가 크면 최소값을 갖는 포인트와 거리가 멀리 떨어져 있어 많이 이동하고, 접선 기울기가 0에 가까울수록 조금씩 움직이도록 한다. Cost(a,b)함수는 식(3)과 같이 열거할 수 있다.

$$Cost=\frac{1}{N}\sum_{i=1}^{n}(ax_i+b-y_i)^2 \quad \cdots (3)$$
$$=\frac{1}{N}\sum_{i=1}^{n}a^2x_i^2+b^2+y_i^2+2ax_ib-2by_i-2ax_iy_i$$

a와 b가 각각 독립변수이므로 a에 관한 미분과 b에 관한 미분을 하면 된다. a의 기울기는 $\frac{\partial Cost(a,b)}{\partial a}=\frac{2}{N}\sum_{i=1}^{n}(ax_i+b-y_i)x_i$, b의 기울기는 $\frac{\partial Cost(a,b)}{\partial b}=\frac{2}{N}\sum_{i=1}^{n}ax_i+b-y_i$ 이며, 편미분하여 합산하면 손실함수의 최소값을 구할 수 있다. 경사 하강법에서 최소 손실을 찾기 위한 기울기는 0이어야 하므로 계속 반복 계산을 통해 최적의 a와 b값을 찾는다.

3.2 스크래핑 탐지 모델 시뮬레이션

R에서 제공하는 선형회귀 함수인 lm()을 사용하여 시뮬레이션한 결과는 그림 2와 같다. 독립변수(x)는 http request 순서를 의미하며, 종속변수(y)는 직전 http request와의 시간 간격을 의미한다. 3~6초 간격을 랜덤하게 만들어 100회의 http request를 요청하였을 경우 로그 데이터를 회귀 분석한 결과 가설함수로 $y=0.0004013x+4.5548132$ 가 도출되었다. 기울기(ⓐ)는 0에 근접한 0.0004013으로 http request가 일정한 시간 간격 범위 안에서 이루어지고 있음을 보여주고 있으며 스크래핑으로 판단할 수 있다. 그리고 그 시간 간격은 y절편값(ⓑ)으로 추정하면 평균 약 4.55초이다. 이는 스크래퍼가 평균 4.55초 간격으로 http request를 보내고 있다고 추측할 수 있다.

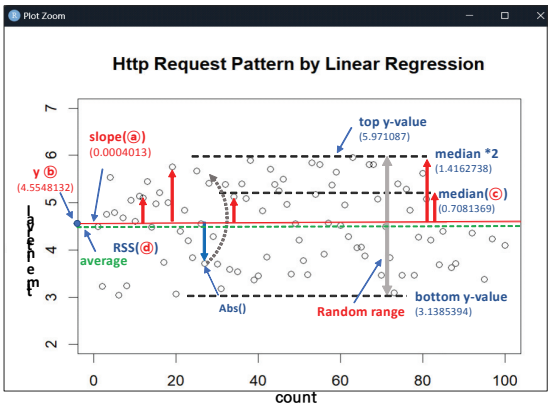

그림 2. 선형회귀를 이용한 http request 패턴 탐지
Fig. 2 Http request pattern by linear regression

고정주기 또는 랜덤주기로 들어오는 http request에 대해 해당 주기의 범위를 추정하는 방법은 우선 y절편인 ⓑ값에서 각각의 y값을 빼고 모두 절대값으로 양수화 시킨다. 그렇게 하면 ⓑ값은 base line이 된다. 그리고 양수화시킨 값들의 중간값을 구하고 base line을 기준으로 중간값의 2배를 상단으로 아래를 하단으로 잡으면 적정한 랜덤값의 범위를 계산할 수 있다. 그림 2의 시뮬레이션 결과에서는 중간값이 0.7081369이며 중간값의 2배인 1.4162738을 ⓑ값에서 빼면 하단 값으로 3.1385394가 나오고 더하여 상단 값을 추정하면 5.971087이 나온다. 이것으로 랜덤값을 3~6 범위로 하고 해당 값만큼 sleep(단위: 초)한 후에 http request를 하는 것으로 추정할 수 있다.

위 선형회귀 분석의 기울기(ⓐ)는 0.0004였다. 그렇다면 기울기가 어느 정도 수준일 때 http request가

일정한 반복 패턴을 띄고 있다고 볼 수 있는지에 대한 판단이 필요하다. 그래서 랜덤 값(시간 간격)의 범위를 1~N초 간격으로 변화시키면서 100회씩 시뮬레이션 후 평균치로 표 1의 결과가 도출되었다.

표 1. 랜덤 구간별 스크래핑 시뮬레이션 결과
Table 1. Scraping simulation output for random intervals

random interval(sec)	slope (ⓐ)	y-intercept (ⓑ)	median (ⓒ)*	RSS average (ⓓ)**
1~3	0.000291	1.988234	0.493594	0.569756
1~6	0.000045	2.994867	0.984413	1.135063
1~10	-0.000001	5.477042	2.248454	2.572817
1~30	0.002341	15.385250	7.125118	8.270454
1~50	-0.002323	25.788110	12.209160	14.005120
1~100	-0.003156	51.038440	24.735310	28.440200

*중간값(ⓒ): 관측값에서 y절편(ⓑ)를 뺀 값의 중간값
**RSS평균(ⓓ): 잔차제곱합 RSS(sum(관측값 - 예측값)^2)을 관측값 개수로 나눈 평균값

시뮬레이션 결과 기울기(ⓐ)는 절대값 0.003 이하 수준일 경우 일반적인 다건 데이터 스크래핑이라 추정할 수 있을 것으로 판단된다. 그러나 랜덤 구간의 길고 짧음에 상관없이 한정된 범위 안에 관측값이 골고루 퍼져있으면 기울기만으로는 스크래핑 여부를 판단하기 어렵다. 그래서 스크래핑 여부 판단과 패턴을 찾기 위해 중간값(ⓒ)이나 RSS 평균값(ⓓ)을 참고해야 한다. 중간값(ⓒ)은 관측값에서 y절편(ⓑ)을 뺀 값들의 median() 한 중간값이다. 중간값과 RSS 평균값 중 시뮬레이션 결과로는 근소하지만 스크래핑 패턴을 알아내기에는 중간값을 사용하는 것이 조금 더 적합하였다. 랜덤 구간별로 y절편 + 중간값*2배 < max(랜덤)이어야 하는데, RSS 평균값을 적용하면 max(랜덤) 값보다 조금 더 높은 수치가 나와 랜덤 범위를 넘어서기 때문이다.

이제 중간값(ⓒ)을 어느 정도로 설정해야 스크래핑으로 판단할지를 정해야 하는데, 보통 스크래핑은 1~10초 사이의 랜덤 값으로 시간 간격을 주어 시도할 것으로 예상한다. 10초를 넘어서면 스크래핑의 기계적인 데이터 수집 효율성이 인력으로 할 경우와 비슷하여 비효율적이기 때문이다. 그러므로 스크래핑의 판단 여부는 시뮬레이션 결과 선형회귀의 기울기(ⓐ)는 절대값으로 0.001 수준 미만이고, 중간값(ⓒ)은 약 2.25보다 작을 경우 스크래핑으로 판단할 수 있을 것이다. 이 경우 중간값은 2.25로 중간값의 2배인 4.5를 상하단의 범위를 잡으면 9초가 되며 이는 이상적 범위(9초, 1~10초 간격)와 같다. 그러나 패턴 범위를 결정하는 중간값은 데이터를 제공하는 각각의 웹사이트 성격에 따라 적정한 수치를 개별적으로 산정해야 할 필요가 있다.

앞서 설계한 스크래핑 모델과 시뮬레이션한 내용을 실제 개발하기 위해 데이터 분석과 시각화를 위한 R과 Shiny 라이브러리, 그리고 실시간 데이터 복제를 위한 Kafka 오픈소스를 사용한다.

3.3 스크래핑 탐지 시스템 아키텍처

Shiny는 웹 어플리케이션 프레임워크로서 웹을 구성하는 html, css를 모르더라도 간편하게 데이터를 시각화하거나 대시보드 형태의 직관적인 화면 개발을 지원하여 R 개발자에게 웹 개발의 부담을 줄여주고 있다. 또한, 원천 로그 데이터를 저장하는 Source DBMS와 로그 데이터를 분석하기 위한 Target DBMS 간의 실시간 데이터 동기화를 위해 Kafka 오픈소스를 사용한다.

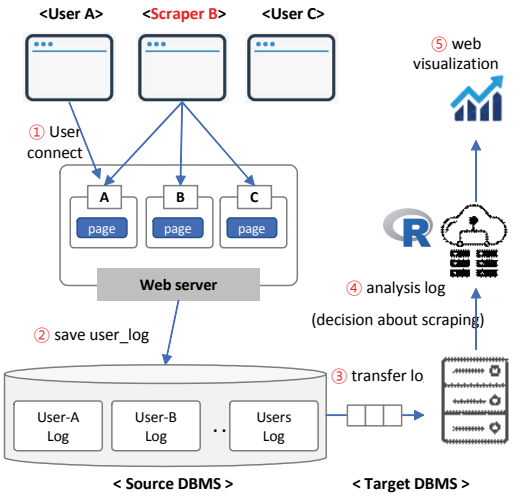

그림 3. 스크래핑 탐지 흐름도
Fig. 3 Flow of scraping detection

실시간 데이터 복제를 반영한 스크래핑 탐지의 전체 흐름은 그림 3과 같다. ① 사용자별로 웹사이트에

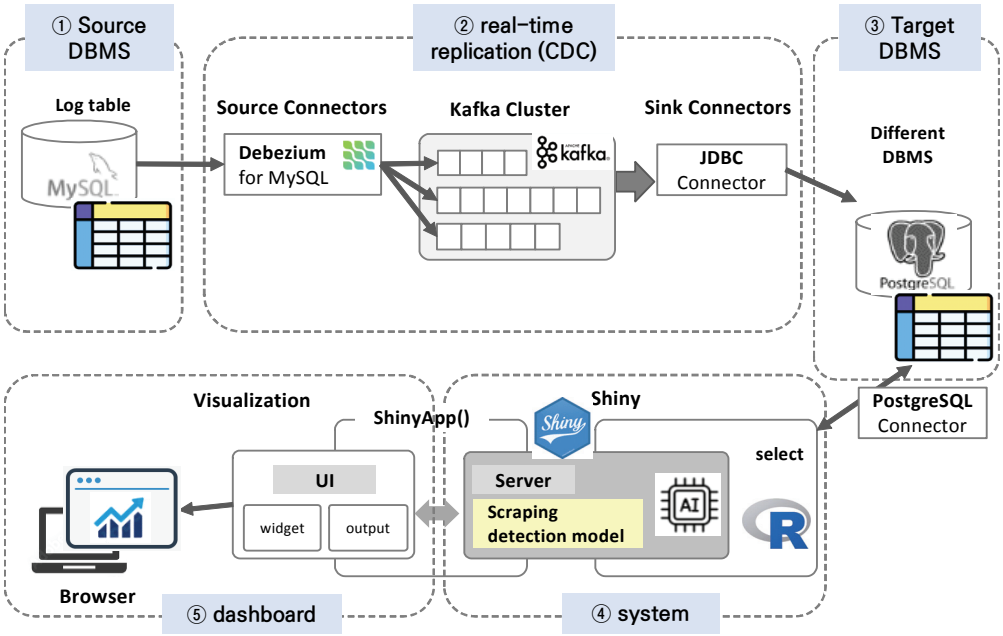

그림 4. 스크래핑 탐지 시스템 아키텍처

Fig. 4 Architecture of scraping detection system

접속하여 데이터 서비스를 이용하고, ② 웹서버는 사용자별 데이터 서비스 이용 내역을 로깅한다. ③ 홈페이지 조회 로그를 실시간으로 분석서버로 전송(복제)후, ④ 분석서버에서 R을 이용하여 스크래핑 여부를 분석하고, ⑤ Shiny를 통해 분석 내용을 웹으로 시각화한다.

사용자 로그를 저장하는 로그 테이블이 있는 Source DBMS에서 분석서버의 Target DBMS까지의 전체적인 시스템 아키텍처는 그림 4와 같다. ① 우선 Source DBMS는 웹 서비스를 통해 제공하는 데이터와 이를 조회하는 사용자의 이용 내역을 기록 저장한다. 본 논문에서는 동일 기종의 replication이 아닌 이기종의 DBMS를 사용하여 Kafka 동작을 검증해 보기 위해 소스는 MySQL을 타겟은 PostgreSQL을 사용한다. ② Kafka의 Debezium Connector를 이용하여 실시간으로 Source DBMS의 DB 로그를 읽어, row 단위로 변경된 데이터를 파싱한 뒤 Kafka의 큐에 담는다. Target DBMS로 데이터를 보내기 위해 Sink Connector을 이용하여 JDBC로 타겟 DBMS와 연결한다. ③ Sink Connector를 통해서 수신한 데이터를 타겟 DBMS의 동일한 테이블에 SQL(Update & Insert) 처리한다. 이로써 소스와 타겟 DBMS의 DB 테이블 간 데이터가 서로 일치하게 된다. ④ 스크래핑 탐지 분석 서버에서 R을 이용하여 로그 데이터를 분석한다. RPostgres 라이브러리를 이용하여 타겟 DBMS에 접속후 복제된 데이터를 읽어 들인다. 이후 앞서 본 논문에서 제안하고 시뮬레이션해 본 선형회귀를 이용한 스크래핑 탐지 모형을 가동시켜 빠르게 분석한다. 탐지 모형은 R로 구현하며 데이터 처리에 편리한 dataframe 사용을 위해 dplyr 라이브러리를 사용한다. ⑤ 분석한 결과와 관련 데이터를 Shiny 라이브러리를 이용하여 시각화하여 웹 화면으로 제공한다.

Ⅳ. 스크래핑 탐지 시스템 구현

4.1 스크래퍼 로그 테이블 설계

스크래핑 여부를 탐지하기 위한 로그 테이블 설계는 표 2와 같으며 사용자의 조회 행위를 저장한다.

표 2. 스크래퍼 로그 테이블 설계

Table 2. Design of scraper logging table

logical name	physical name	format	length
base date	bas_dt	date	-
base time	bas_tm	time	-
Timestamp	bas_ts	timestamp	-
user ID	user_id	varchar	30
page code	scrn_cd	char	10
inquiry target	iq_trgt_bzno	char	10
IP address	ip_addr	varchar	30

4.2 분석용 DB로 실시간 데이터 복제

설계한 로그 저장 DB 테이블을 분석용 DB로 실시간 전송(복제)을 하기 위해 Kafka Debizium Source Connector를 설정한다. 그림 5는 MySQL을 소스 DB로 하여 REST-API로 Connect를 생성하는 내용이다.

```
curl --location -request POST
'http://kafka-server-ip:8083/connectors'
--header 'Content-Type: application/json'
--data-raw '{
  "name": "source-connector-mysql-db1",
  "config": {
    "io.debezium.connector.mysql.MySqlConnector",
    "database.hostname": "src-db
    "database.port": "3306",
    "database.user": "cdc",
    "database.password": "password123",
    "database.include.list": "db1",
    "topic.prefix" : "cdcdb1",
"schema.history.internal.kafka.topic":"cdc-db1
    -- midterm omission --   } }'
```

그림 5. Debezium source connector 생성

Fig. 5 Creation of debezium source connector

Kafka에서 소스 DB의 로그를 topic이라 부르는 row 단위로 데이터를 가져오면, 이제 타겟 DB로 데이터를 보내야 한다. 타겟 DB용 Connector 설정도 REST-API로 그림 6과 같이 생성할 수 있다.

4.3 스크래핑 탐지 분석 모형 구현

소스 DB에서 타겟 DB로 데이터가 실시간 복제되고 탐지 모형 프로그램이 주기적으로 타겟 DB를 이용하여 스크래핑 여부를 분석한다. 스크래핑 탐지 구현 흐름도는 그림 7과 같다.

```
curl --location -request POST
'http://kafka-server-ip:8083/connectors'
--header 'Content-Type: application/json'
--data-raw '{
  "name": "sink-connector-postgres-db1",
  "config": {
   "io.confluent.connect.jdbc.JdbcSinkConnector",
   "connection.url":
"jdbc:postgresql://target-db:5432/sinkdb?currentSc
hema=db1",
    "connection.user": "debezium_postgres",
    "connection.password":
                "debezium_postgres123",
    "table.name.format":"${topic}",
    -- midterm omission --   } }'
```

그림 6. Debezium target connector 생성

Fig. 6 Creation of debezium target connector

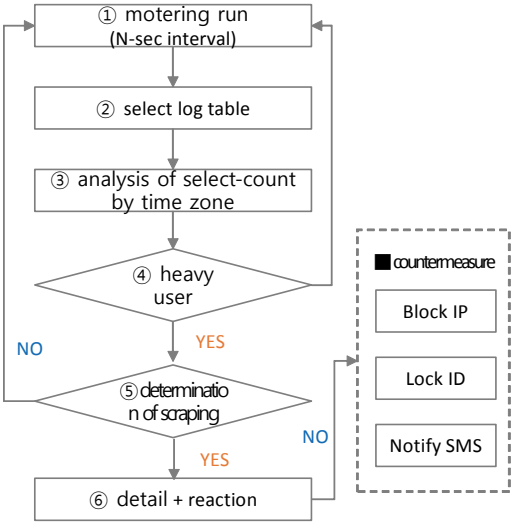

그림 7. 스크래핑 탐지 모형 흐름도

Fig. 7 Flow of scraping detection model

① Shiny로 개발한 웹 모니터링 화면은 기본으로 5초 간격으로 타겟 DB에서 현재 날짜 시점의 분석할 데이터를 조회한다. ② 조회할 데이터는 사용자가 화면에서 Calendar 컴포넌트로 날짜를 변경할 수 있다. ③ 해당 날짜의 로그를 읽어 시간대별로 사용자의 조회 횟수를 시각화하여 보여준다. ④ 일정 횟수 이상으로 조회하는 사용자가 있는지 확인한다. ⑤ 해당 사용자가 스크래핑 중인지를 탐지 모형으로 분석한다. ⑥

스크래핑 중으로 분석되면 해당 내용을 모니터링 화면에 출력하고 대응 조치를 한다. 데이터 유출 방지를 위한 대응 조치로는 스크래퍼의 IP를 서비스 안 되도록 차단하거나, 사용자 ID를 잠금 처리하여 로그인을 못하도록 하고 운영 담당자에게 SMS 메시지를 발송하여 현재 상황을 통보하는 방법이 있을 수 있다[10].

R로 구현한 로직에 대해 핵심적인 부분만 설명하면, 로그 테이블에서 시간대별로 결합하여 사용자 조회 타임스탬프 및 건수를 dataframe으로 만든다. 이후 타임스탬프의 앞뒤 시간 간격을 모두 계산한 dataframe을 만들어 선형회귀 모형의 입력으로 사용한다. 그림 8은 선형회귀 모델 처리 내용이다.

```
# dataset for data analysis
scrpdf = subset(logdataata, iq_who == top_iq_who,
        select= c("hours", "ts", "scrn_cd"))
TS = subset(scrpdf, select= c("ts"))
rows = nrow(TS)
x = TS[1:rows-1,]   # time Interval
y = TS[2:rows,]
# time interval dataframe for linear regression
modeldf = data.frame(step= seq(from=1,
     to=rows-1), timeeInterval= as.numeric(
     difftime(y, x, tz = "Asia/Seoul",
     units = "secs")))
# run linear regression model
model= lm(data=modeldf, timeInterval~step)
a = model$coefficients['step'] # slope
b = model$coefficients['(Intercept)'] # y-intercept
Rsquared = summary(model)$r.squared
# Interval
timeIntervalTerm = modeldf$timeInterval - b
# convert abs()
timeIntervalTermAbs = abs(timeIntervalTerm)
timeIntervalTermMed = median(timeIntervalTermAbs)
# median value * 2 (range of random-value)
randomRange = timeIntervalTermMed * 2
```

그림 8. 선형회귀 처리

Fig. 8 Linear Regression model processing

4.4 분석 결과 시각화

그림 9는 스크래퍼의 로그 정보를 목록화하여 나타낸 것으로 스크래핑 시도한 시간대와 각 조회 시간의 타임스탬프, 사용자 ID, 조회화면 코드, 조회 대상, 사용자 IP를 보여준다.

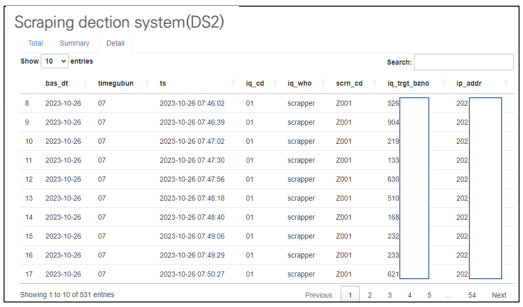

그림 9. 모니터링 로그 상세 목록

Fig. 9 Detail list of monitoring log data

그림 10은 스크래핑 탐지 분석 결과에 대한 화면이다. x축은 조회 순차값을 의미하며, y축은 조회하고 나서 다음 조회한 시간까지의 시간 간격(interval)을 의미한다. 붉은색은 선형회귀를 나타낸 가설함수 라인으로 기울기가 0에 가까운 모습을 보이고 있으며 스크래핑을 18~29초 범위 안에서 랜덤한 시간 동안 sleep 하다가 다시 스크래핑을 시도하는 모습으로 판단할 수 있다.

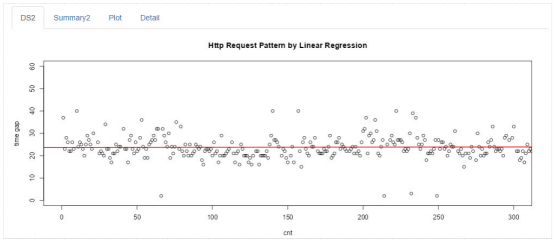

그림 10. 스크래핑 패턴 그래프

Fig. 10 Graph of scraping pattern

Ⅴ. 결론

본 논문에서 제안하는 스크래핑 탐지 모형은 웹사이트에 로그인한 사용자가 비공개성 데이터를 조회한 시점의 타임스탬프를 로깅하고 기록한 타임 스탬프 간의 시간 간격을 이용한다. 이를 분석 대상으로 하여 반복적으로 시도되는 데이터 조회 요청의 시간 패턴을 알 수 있다. 탐지 모형으로는 인과관계를 갖는 두 변량 사이의 함수관계를 통계적으로 분석하여 예측하

는 선형회귀를 적용하였다. 선형회귀의 제1 변량은 http request 순번이고 제2 변량은 각 request 간의 타임스탬프 시간 간격이다. 정해진 시간 범위 안에서 랜덤한 간격으로 http request를 하는 스크래핑은 일정한 시간적 패턴이 있으며 시뮬레이션 결과 스크래핑 시도 횟수가 많을수록 선형회귀의 회귀계수(기울기)는 0에 가깝게 수렴하였다. 0으로의 수렴은 관측값이 가로축으로 일렬인 형태이며 이는 시간 측면에서 유사한 반복 패턴이 있는 스크래핑으로 판단할 수 있다. 보통 1~10초 간격의 현실적인 스크래핑 시도를 하게 되는 경우 회귀계수는 0.001 미만으로 도출되었다. 또한, 선형회귀 분석을 통해서 랜덤한 시간으로 스크래핑을 시도하더라도 그 시간 간격의 범위를 유추해 낼 수 있었으며 그 범위 값은 RSS 평균값보다 중간값으로 유추하는 것이 약 14% 더 정확한 것으로 나타났다.

향후 과제로는 제안한 방법은 일정 시간 범위 안에서 랜덤한 시간 간격으로 시도하는 스크래핑의 경우는 탐지할 수 있으나, 시간 범위 구간을 여러 개로 다양하게 설정하고, 구간을 변경해 가며 스크래핑을 시도하는 경우는 선형회귀 형태가 되지 못할 경우가 높아 탐지하기가 어려워진다. 이를 해결하기 위해 입력 데이터를 선처리하여 군집화 작업을 하고 군집한 데이터 안에서 패턴을 찾는 식의 추가 연구가 필요하다.

감사의 글

본 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2016R1D1A1B02008553). 본 논문은 교육부 및 한국연구재단의 4단계 BK21 사업(경북대학교 컴퓨터학부 지능융합 소프트웨어 교육연구단)으로 지원된 연구임(4120240214871)

References

[1] Ministry of Science and ICT, “2024 digital new-deal action plan – Digital new-deal main achievements and future analysis,” *Report,* Jan. 2022, pp. 5-7.

[2] National Assembly of The Republic of Korea, “Act on the promotion of data industry and the activation of data use,” *Report,* Apr. 2022, pp. 2-6.

[3] J. Lee, "The legislative meaning and limitations of the Act on the promotion of data industry and the activation of data use,” *Study of IT and Law*, vol. 24, no. 2, 2022, pp. 265-296. https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11266035

[4] W. Jung, J. Kim, and N. Park, "Web-browing application using web scraping technology in korean network separation application,” *Symmetry*, vol. 13, no. 8, July 2021, pp. 1-17. https://doi.org/10.3390/sym13081550

[5] D. Sasaki and A. Iwasaki, *Apache Kafaka: Construction and Activation of Distributed Messaging Systems.* Seoul: Hanbit Publishing Network, 2020.

[6] K. Medeiros, *R Programming Fundamentals.* Birmingham: Packt Publishing, Sept. 2018.

[7] C. Beeley, *Web Application Development with R Using Shiny.* Birmingham: Packt Publishing, Jan. 2016.

[8] Stripters Inc, “Method for Detecting Web Scraping and Server for Executing the Same,” *Korean Patent,* no. 102595303, Apr. 2021.

[9] E. Lee, G. Bak, J. Lee, and Y. Bae, "Prediction of budget prices in electronic bidding using deep learning model," *J. of the Korea Institute of Electronic Communication Sciences*, vol, 18, no, 6, 2023, pp. 1171-1176. https://doi.org/10.13067/JKIECS.2023.18.6.1171

[10] I. Hwang, "The influence of information security policy, technology, and communication uncertainties: The role of information security role identity," *J. of the Korea Institute of Electronic Communication Sciences*, vol, 19, no, 1, 2023, pp. 241-248. https://doi.org/10.13067/JKIECS.2024.19.1.241

저자 소개

이지율(Ji-Yul Lee)

2003년 강원대학교 정보통신공학과 졸업(공학사)
2024년 경북대학교 대학원 컴퓨터학부 계약학과 졸업(공학석사)

2003년~현재 신용보증기금 ICT 직군 근무
※ 관심분야 : 빅데이터, 시스템 아키텍처

이용주(Yong-Ju Lee)

1985년 한국과학기술원 정보검색전공(공학석사)
1997년 한국과학기술원 컴퓨터공학전공(공학박사)

1998년 8월 ~ 현재 경북대학교 IT대학 컴퓨터학부 교수
※ 관심분야 : 시맨틱 웹, 빅데이터, 딥러닝