

논문 2026-3-5 <http://dx.doi.org/10.29056/jsf.2026.09.05>

불법 스트리밍 사이트의 DOM 변형에 대한 요소 로케이터 유형별 강인성 비교

장의진*, 이지은**, 임형민***†

Comparative Robustness of Element Locator Types under DOM Transformations in Illegal Streaming Sites

Uijin Jang*, Jieun Lee**, Hyungmin Lim***†

요 약

불법 스트리밍 사이트는 차단을 회피하기 위해 도메인과 마크업 구조를 반복적으로 변경하며, 이는 침해 채증에 사용되는 XPath와 CSS 선택터를 무력화한다. 선행 연구는 변형 로그로 정답 로케이터를 복구하는 방법을 제시하였으나, 실제 추적 환경에서는 그러한 로그가 제공되지 않는다. 본 논문은 선행 연구의 변형 규칙에 노드 삭제와 텍스트 변형을 추가하고 세 단계의 상대적 스트레스 수준을 도입하여 변형 연산자를 다섯 영역으로 재구성하였으며, 공개 아카이브에서 확보한 13개 브랜드 21개 페이지의 63개 대상에 대해 여섯 가지 요소 지정 방식을 측정하였다. 절대 XPath는 모든 대상에 적용되나 가장 약한 스트레스 수준에서 이미 생존율이 20.3%로 급락하였고, ID 선택터는 변형에 강인하나 생성률이 22.2%에 그쳐 강인성과 적용 가능성이 상충하였다. 또한 변형 영역마다 무력화되는 지정 방식이 서로 달라 여러 방식을 병행하는 접근이 유효할 수 있음을 확인하였다. 본 결과는 침해 대응 수집 시스템의 로케이터 설계 지침으로 활용될 수 있다.

Abstract

Illegal streaming sites repeatedly change their domains and markup structure to evade blocking, invalidating the XPath and CSS selectors used to collect infringement evidence. A prior study restored ground-truth locators from a mutation log, which is unavailable when tracking real sites. We extend the prior mutation rules with node deletion and text transformation, introduce three relative stress levels, reorganize the operators into five areas, and evaluate six element location strategies on 63 targets from 21 pages of 13 site brands in public web archives. Absolute XPath applies to all targets, yet its survival rate drops to 20.3% even under the weakest level, whereas ID selectors are robust but can be generated for only 22.2% of targets, revealing a trade-off between robustness and applicability. Since different transformation areas invalidate different strategies, combining multiple strategies may be effective. These findings can inform locator design in infringement-monitoring collection systems.

한글키워드 : 불법 스트리밍 사이트, DOM 구조 변형, 요소 로케이터, 강인성 측정, 저작권 침해 대응

keywords : illegal streaming site, DOM transformation, element locator, robustness measurement, copyright infringement response

* 숭실대학교 스파르탄AI교육원

** 숭실대학교 컴퓨터학과

*** ㈜엘엔제이테크

† 교신저자: 임형민(email: atskyo@gmail.com)

접수일자: 2026.08.12. 심사완료: 2026.08.19.

게재확정: 2026.09.20.

1. 서론

온라인 영상 유통이 확대되면서 정식 유통 권한을 확보하지 않은 불법 스트리밍 사이트를 통한 콘텐츠 재송출 문제가 지속되고 있다. 국내에서는 스트리밍 링크 사이트를 통한 불법 이용이 불법복제물 이용량의 다수를 차지하며, 해당 경로의 이용량은 2025년 전년 대비 23.4% 증가하였다[1]. 해외에서도 2024년 기준 접속 가능한 한류콘텐츠 불법유통 사이트가 17,119개로 확인되었으며, 영상 분야는 주로 스트리밍 링크 사이트 형태로 운영되는 것으로 나타났다[2]. 이에 대응하기 위해 각국은 사이트 차단 제도를 운영해 왔으며, 최근에는 차단 절차를 단축하거나 이동한 도메인까지 포괄하여 차단할 수 있도록 제도가 개편되고 있다[3]. 국내에서는 2026년 저작권법 개정으로 한국저작권보호원의 접속차단 권고 권한과 문화체육관광부 장관의 긴급차단 명령 제도가 도입되었고, 같은 해 5월 제도 시행 첫날 34개 사이트에 대한 긴급차단 명령이 통지되었다[4].

이러한 제도 변화는 불법 스트리밍 사이트의 대응 양상에도 영향을 미친다. 차단 주기가 짧아질수록 운영자는 도메인을 더 빈번히 교체하고 대체 사이트를 개설하며, 이 과정에서 화면 구성과 마크업 구조도 함께 변경된다. 실제로 동일한 사이트가 URL과 도메인을 계속 바꾸며 차단을 회피하는 양상이 보고되고 있고[3], 2020년에 발견된 불법유통 사이트 중 2024년까지 접속이 유지된 것은 절반에 미치지 못하였다[2]. 문화체육관광부 역시 긴급차단 시행 이후 대체 사이트 재생성 추이를 지속 관찰하며 차단 대상을 확대할 계획임을 밝힌 바 있다[4]. 본 논문의 사전 조사에서도 동일한 서비스명을 사용하는 사이트가 최대 6개의 서로 다른 도메인으로 운영되고 있음이 확인되었다.

차단 명령의 실효성은 대상 사이트를 지속적으

로 식별하고 침해 사실을 입증할 수 있는지에 달려 있다. 2026년 개정 저작권법이 도입한 긴급차단 명령은 신속한 조치를 가능하게 하였으나, 명령의 근거가 되는 침해 채증은 여전히 대상 사이트를 반복 방문하여 콘텐츠를 확인하는 절차에 의존한다. 수집 자동화가 중단되면 채증 주기가 길어지고 차단 대상의 갱신이 지연된다. 따라서 로케이터의 강인성은 기술적 문제인 동시에 제도 운영의 실효성과 연결된 문제다.

불법 스트리밍 대응 연구는 크게 두 방향으로 진행되어 왔다. 하나는 수집된 영상에 대한 특징 추출과 식별에 관한 연구이고[5], [6], 다른 하나는 사이트 자체의 정보를 수집·추적하는 연구이다[7] - [9]. 전자의 경우 영상 변형 유형을 분류하고 강인성 평가 기준을 제시하거나[5], 입력 영상을 식별에 적합한 형태로 정규화하는 방법[6]이 제안되었다. 그러나 이들 연구는 질의 영상이 이미 확보된 상태를 전제로 하며, 영상을 확보하기 이전 단계인 사이트 접근 및 수집 과정의 변형은 다루어지지 않았다.

저작권 보호 기관과 침해 대응 시스템은 불법 스트리밍 사이트를 주기적으로 순회하며 유통 중인 콘텐츠를 확인하고 침해 증거를 확보한다[7] - [9]. 대상 사이트가 많고 도메인이 자주 바뀌는 환경에서는 이 과정의 자동화가 전제된다. 자동 수집은 재생 화면, 콘텐츠 목록, 제목과 같은 화면 요소의 위치를 지정하는 것에서 시작하며, 이때 XPath나 CSS 셀렉터와 같은 요소 로케이터가 사용된다. 이러한 로케이터는 대상 페이지의 마크업 구조가 변경되면 대상을 찾지 못하게 되며, 이는 웹 테스트 자동화 분야에서 로케이터 취약성 문제로 오랫동안 다루어져 왔다[11], [12]. 최근에는 이 문제를 불법 스트리밍 사이트 추적 환경에 적용한 연구[10]가 제시되어, 변형 규칙을 이용한 데이터 증강과 변형 로그 기반의 정답 XPath 복구 방법을 제안하였다.

그러나 실제 추적 환경에서는 사이트 구조 변경에 대한 변형 로그가 제공되지 않는다. 따라서 복구 기제 없이 기존 로케이터가 어느 정도까지 유지되는가, 그리고 어떤 지정 방식이 어떤 변형에 취약한가는 별도로 확인해야 한다.

본 논문에서는 변형 로그가 주어지지 않는 조건에서 요소 지정 방식의 강인성을 비교 측정한다. 측정은 탐색 대상 지정, 변형 연산자 적용, 로케이터 생성 및 해석, 생성물과 생존율 산출, 페이지 단위 통계 검정으로 구성되며, 비교 대상은 절대 XPath, ID 선택터, class 선택터, 상대 XPath, 텍스트, class 토큰의 여섯 가지이다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 검토한다. 제3장에서는 변형 연산자와 강도 기준, 비교 대상 요소 지정 방식을 기술한다. 제4장에서는 실험 환경과 측정 결과를 분석하고, 제5장에서 결론과 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 불법 스트리밍 영상의 변형과 식별

불법 스트리밍 영상은 원본과 동일한 내용을 포함하더라도 수집·변환·재송출 과정에서 다양한 왜곡을 포함한다. 재인코딩에 따른 화질 저하, 해상도 및 화면비 변환, 밝기·색상 변화, 회전과 반전, 프레임율 변화 등이 대표적이며, 화면 녹화나 재송출 과정에서 검은 여백, 자막, 로고, 광고 영역과 같은 비콘텐츠 요소가 포함되기도 한다[5], [6].

기존 연구에서는 불법 스트리밍 환경에서 나타나는 영상 변형 유형을 시각적 변형, 구조적 편집 변형, 서비스 인터페이스 중첩, 실시간 스트리밍 환경 변형, 획득·재전송 환경 변형, 복합 변형의 6개 영역으로 구분하고, 각 영역에 대한 세부 평가 항목과 파라미터, Level 1~3의 수준 구분 기준을 제시하였다[5]. 특히 플레이어 UI, 광

고 배너, 사이트 로고와 같이 원본 영상에는 존재하지 않으나 실제 탐지 대상 영상에는 포함되는 요소를 별도 평가 항목으로 다루었다는 점에서 기존 복제 탐지 평가와 차별성을 가진다.

또한 유효 영역 및 프레임 기반 입력 영상 정규화 방법이 제안되었다[6]. 이 방법은 유효 영역 검출, 종횡비 보존 해상도 정규화, 정보량 기반 프레임 정제, 회전 후보 기반 방향 보정, 유효 프레임 선별로 구성되며, 정규화 미적용 조건 대비 식별 인식률이 향상됨을 보였다.

이들 연구는 확보된 영상에 대한 처리를 다루며, 영상이 어떤 경로로 수집되었는지는 전제 조건으로 둔다. 본 논문은 그 전제 조건에 해당하는 수집 단계를 대상으로 한다는 점에서 위 연구들과 상보적인 관계에 있다. 특히 픽셀 계층에서 서비스 인터페이스 중첩으로 분류되는 플레이어 UI, 광고 배너, 사이트 로고는 DOM 계층에서는 위치와 속성이 명시된 노드로 존재한다. 동일한 현상을 픽셀에서는 제거 대상 잡음으로 다루는 반면, 본 논문에서는 구조 변형을 일으키는 삽입 노드로 다룬다. 즉 두 접근은 같은 대상을 상이한 계층에서 관측한다.

2.2 불법 스트리밍 사이트 정보 수집 연구

IP 포트 스캐닝을 이용한 불법 유통 플랫폼 추적 방법[7], 광고주 고유 식별번호를 활용한 저작권 침해 사이트 특징 정보 수집 전략[8], 텍스트 수치화에 기반한 불법 스트리밍 사이트 구조 분석 방법[9] 등 사이트 자체를 대상으로 하는 연구도 진행되어 왔다.

이 중 [9]는 사이트의 구조적 특성을 분석 대상으로 삼는다는 점에서 본 논문과 인접한 연구이다. 다만 [9]가 사이트 간 유사성 판별과 분류를 목적으로 페이지의 텍스트 정보를 수치화하는데 초점을 두는 반면, 본 논문은 동일 사이트의 구조가 변경될 때 특정 화면 요소를 계속 지정할

수 있는가를 다룬다. 즉 [9]가 사이트 단위의 식별을 다룬다면 본 논문은 요소 단위의 추적 지속성을 다루며, 평가 대상과 지표가 구분된다.

2.3 웹 요소 로케이터의 강인성 연구

웹 페이지에서 특정 요소를 지정하는 로케이터는 웹 테스트 자동화와 데이터 수집에서 공통적으로 사용된다. 절대 XPath는 문서 루트부터의 경로를 그대로 기술하므로 중간 노드가 추가되거나 순서가 바뀌면 즉시 무효가 되고, ID나 class 기반 선택터는 경로 변화에는 견디지만 속성값이 변경되면 대상을 찾지 못한다. 이 문제를 완화하기 위한 연구는 크게 세 갈래로 진행되어 왔다.

첫째는 변경에 견디는 로케이터를 자동으로 생성하는 접근이다. Leotta 등[11]은 페이지 변경에도 대상 요소를 계속 찾을 가능성이 높은 XPath를 생성하는 ROBULA+를 제안하였으며, 절대 XPath 및 Selenium IDE 방식보다 로케이터 취약성을 유의미하게 줄였다고 보고하였다. 이후 통계적 정보를 활용하여 로케이터를 적응적으로 생성하는 SIDEREAL[12]이 제안되었고, Nguyen 등[13]은 대상 요소를 이웃 요소와의 의미적 구조 관계를 이용하여 지정하는 기법을 제안하여 15개 웹사이트를 대상으로 한 실험에서 Selenium IDE 및 ROBULA+보다 잘못된 요소를 반환하는 경우가 적었음을 보였다.

둘째는 유사도를 이용하여 대상을 탐색하는 접근이다. Nass 등[14]은 단일 속성이 아니라 태그, 속성, 텍스트, 화면상 위치 등 다수의 정보를 종합한 유사도로 요소를 찾는 Similo를 제안하였다. 이후 여러 후보 중 최적을 선택하기 위한 순위화 기법[15]과, 대규모 언어 모델을 활용하여 실패한 로케이터를 보완하는 방법[16]으로 확장되었다. 또한 화면상 겹침을 고려하여 요소 식별의 강인성을 높이는 방법[19]이 제안되었다.

셋째는 로케이터가 깨지는 원인을 분석하고

복구하는 접근이다. Nass 등[17]은 GUI 테스트 자동화에서 발생하는 문제를 실증적으로 분석하여 요소 식별 실패가 주요 원인 중 하나임을 보였고, Kirinuki 등[18]은 DOM의 속성과 구조에만 의존하지 않고 자연어 정보를 함께 이용하여 깨진 로케이터를 복구하는 방법을 제안하였다.

이들 연구는 기능 개선이나 리팩터링 등 정상적인 개발 과정에서 발생하는 변경을 전제한다. 최근에는 오픈소스 프로젝트에서 수집한 중단간 웹 테스트 데이터셋[20]이 공개되어, 로케이터 파손을 실제 코드베이스 기준으로 분석할 수 있는 기반이 마련되었다.

2.4 불법 스트리밍 사이트 추적을 위한 로케이터 연구

Lee 등[10]은 불법 OTT 스트리밍 사이트 추적 환경의 로케이터 파손 문제를 다루었다. 프론트엔드 구조가 바뀌어도 핵심 기능의 논리 구조는 유지된다는 점에 근거하여 검색 입력과 검색 실행 요소를 기능 중심 앵커로 정의하고, 추적 회피 기법을 모사하는 10개의 변형 규칙과 변형 로그 기반의 적응형 XPath 라벨링을 제안하였다. 20개 사이트에서 생성한 200건의 증강 데이터에서 원본 XPath를 그대로 재사용한 경우 성공률이 94.4%로 저하되었으나, 제안 방법으로 100%까지 회복됨을 보였다.

이 연구는 DOM 변형으로 로케이터가 실제 파손됨을 정량적으로 확인하였다는 점에서 본 논문과 직접 관련되나, 다루는 문제가 다음과 같이 구분된다. 적응형 라벨링은 변형 로그가 존재하는 상황을 전제하지만, 실제 추적 환경에서는 사이트 운영자가 구조를 변경하므로 그러한 정보가 주어지지 않는다. 또한 [10]은 수작업으로 지정한 XPath 두 개만을 정답으로 삼아 지정 방식 간 비교와 생성 가능 여부를 다루지 않았으며, 앵커로 삼은 두 요소는 폼 요소로서 식별 속성이 부여될 가능성이 상대적으로 높다.

본 논문은 복구 기제가 없는 조건에서 여섯 가지 지정 방식을 비교하고 생성물을 별도 지표로 도입하며, 콘텐츠 항목 목록·제목·다음 항목 링크로 대상을 확장한다. 변형 연산자는 [10]의 10개 규칙에 강도 수준과 노드 삭제, 텍스트 변형을 추가하여 확장한 것이며, 구체적인 대응 관계는 3.1절에서 기술한다.

3. 변형 연산자와 요소 지정 방식

3.1 변형 연산자 도출

불법 스트리밍 사이트는 차단·추적에 대응하기 위해 도메인뿐 아니라 화면 구성 요소의 마크업 구조를 반복적으로 변경한다. 본 논문에서는 선행 연구 [10]가 설계한 10개의 변형 규칙을 변형 발생 위치와 로케이터에 미치는 영향에 따라 5개 영역으로 재구성하고 강도 파라미터를 도입하였으며, 그 대응 관계는 표 1과 같다. 확장(대상)은 적용 범위를 넓힌 경우, 확장(파라미터)은 강도 조절 변수를 도입한 경우, 확장(세분)은 삽입 유형을 구분한 경우를 의미한다.

신규 연산자는 실제 사이트에서 관측되는 변형 양상을 반영하여 추가하였다. A3과 A4는 빌드 도구 변경이나 프레임워크·트래킹 스크립트에 의해 발생하고, B5는 목록 구조 재설계에 해당한다. D2~D4는 다국어 운영과 문자열 매칭 회피를 반영한다. 특히 C4(노드 삭제)는 [10]이 노드 삽입만을 다루어 관측되지 않았던 연산자로, 삭제로 인한 인덱스 이동이 경로 기반 로케이터에 미치는 영향을 확인하기 위해 추가하였다.

[10]의 10개 규칙 중 이벤트 핸들러 변형(Event Handler Variation)은 본 논문에서 다루지 않는다. onclick 등 이벤트 바인딩 방식의 변경은 DOM 구조를 바꾸지 않아 구조 기반 로케이터에 영향을 주지 않기 때문이다.

변형 강도를 세 수준으로 구분하는 방식은 영상 계층에서 변형 강인성 평가 기준을 제시한

[5]의 구성을 DOM 계층에 적용한 것이다.

표 1. 선행 연구[10]의 변형 규칙과 본 논문 연산자의 대응 관계

Table 1. Mapping between mutation rules in [10] and the proposed operators

연산자	선행 연구 [10]의 대응 규칙	관계
A1. 클래스명 변경	Attribute Value Obfuscation	동일
A2. id 변경·제거	Parameter Key Substitution	확장(대상)
A3. 속성 순서 변경	-	신규
A4. 무의미 속성 추가	-	신규
A5. 클래스명 난독화	Attribute Value Obfuscation	동일
B1. 래퍼 노드 삽입	Structure Injection (Wrapper)	확장(파라미터)
B2. 태그 치환	Input Tag / Tag Substitution	동일
B3. 형제 순서 재배치	DOM Hierarchy Reordering	확장(파라미터)
B4. 노드 깊이 변경	DOM Hierarchy Reordering	확장(파라미터)
B5. 컨테이너 분할·병합	-	신규
C1. 광고 노드 삽입	Noise Tag Injection	확장(세분)
C2. 오버레이 삽입	Noise Tag Injection	확장(세분)
C3. 장식 노드 삽입	Noise Tag Injection	확장(세분)
C4. 비필수 노드 삭제	-	신규
D1. 레이블 문자열 변경	Placeholder Variation	확장(대상)
D2. 언어 전환	-	신규
D3. 텍스트→아이콘 대체	-	신규
D4. 유사문자 치환	-	신규

※ [10]의 10개 규칙 중 이벤트 핸들러 변형을 제외한 나머지는 위 연산자에 대응된다.

재구성한 5개 영역은 속성 변형(A), 구조 변형(B), 노드 삽입·삭제(C), 텍스트·레이블 변형(D),

복합 변형(E)이며, 각 영역과 해당 영역이 주로 무력화하는 로케이터는 그림 1과 같다.

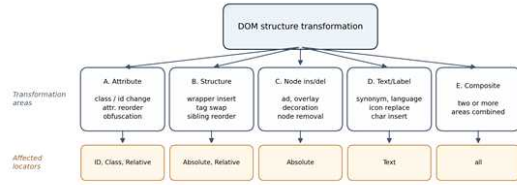


그림 1. DOM 구조 변형 유형 분류 체계와 영향을 받는 로케이터

Fig. 1. Taxonomy of DOM structure transformations and affected locators

3.2 탐색 대상 요소

변형 실험은 무엇을 찾으려 하는가가 고정되어야 성립한다. 불법 스트리밍 사이트 추적에서 실제로 필요한 요소를 표 2와 같이 5종으로 한정한다.

표 2. 탐색 대상 요소
Table 2. Target elements

ID	요소	기능적 정의	수집 목적
T1	플레이어 컨테이너	영상 재생을 담당하는 노드	스트림 엔드 포인트 추출
T2	콘텐츠 항목 목록	개별 작품 또는 회차로 이동하는 링크 집합	수집 진입점 확보
T3	콘텐츠 제목	작품명을 나타내는 텍스트 노드	저작물 식별
T4	검색 입력	사이트 내 검색 폼	탐색 자동화
T5	다음 항목 링크	순차 이동 앵커	연속 수집

[10]이 앵커로 정의한 검색 입력 요소는 본 논문의 T4에 해당한다. 본 논문은 여기에 콘텐츠와 직접 관련된 T1~T3, T5를 추가하였다.

3.3 변형 연산의 형식적 정의

문서 객체 모델(DOM) 트리를 $D = (V, E)$ 라 하고, 탐색 대상 요소의 집합을 식 (1)과 같이 정의한다.

$$D = (V, E), T = \{t_1, t_2, \dots, t_m\} \subset V \quad (1)$$

여기서 V 는 노드 집합, E 는 부모 - 자식 관계의 집합이며, t_i 는 3.2절에서 정의한 탐색 대상에 해당하는 노드이다.

변형 강도 수준 $l \in 0, 1, 2, 3$ 에 대한 파라미터 벡터를 θ_l , 난수 시드를 s 라 할 때, 영역별 변형 연산자를 $\Phi_A, \Phi_B, \Phi_C, \Phi_D$ 로 표기한다. 전체 변형은 식 (2), (3)과 같이 이들의 합성으로 정의한다.

$$\Phi^{(l,s)} = \Phi_D^{(l,s)} \circ \Phi_C^{(l,s)} \circ \Phi_B^{(l,s)} \circ \Phi_A^{(l,s)} \quad (2)$$

$$D^{(l,s)} = \Phi^{(l,s)}(D) \quad (3)$$

복합 변형(영역 E)은 별도의 연산자가 아니라 식 (2)에서 둘 이상의 영역이 동시에 적용되는 경우를 의미한다. 영역별 단독 적용 실험에서는 해당 영역을 제외한 나머지 연산자의 파라미터를 0으로 두어 항등 연산이 되도록 한다.

변형 전후의 노드 대응 관계를 $\mu: V \rightarrow V^{(l,s)} \cup \perp$ 라 하면, 변형이 실험을 무의미하게 만들지 않도록 식 (4)의 불변조건을 강제한다.

$$\forall t \in T: \mu(t) \neq \perp \wedge \text{tag}(\mu(t)) = \text{tag}(t) \quad (4)$$

즉 모든 탐색 대상은 변형 후에도 소멸하지 않으며, 태그로 표현되는 기능적 정체성이 보존된다. 이 밖에 HTML 파싱 가능 상태 유지, 시드 고정에 의한 재현성, 대상 노드 수의 일치를 함께 검사하며, 구현에서는 매 시행마다 자동으로 검사한다.

3.4 변형 유형별 세부 항목 및 강도 수준

각 영역의 세부 항목과 수준별 파라미터는 표 3과 같다. 본 논문의 Lv1~Lv3는 실제 사이트에서 관측된 변경 빈도나 정도를 측정할 값이 아니라, 로케이터의 내성 한계를 확인하기 위해 저자가 설정한 상대적 스트레스 수준이다. 변형을 적용하지 않은 원본 상태는 Lv0으로 표기하며, 로케이터 생성이 올바른지 확인하는 기준 조건으로 사용한다. 파라미터는 다음 세 원칙에 따라 설정하였다.

첫째, 각 연산의 상한(Lv3)은 해당 조작이 논리적으로 도달할 수 있는 최대치로 두었다. 클래스명과 id의 전면 변경, 형제 노드의 전역 재배치, 텍스트의 전면 교체가 이에 해당하며, 원본과의 대응 단서가 모두 소멸하는 극단 조건을 확보하기 위한 것이다.

둘째, 하한(Lv1)은 대부분의 지정 방식이 생존하는 수준에서 시작하도록 두었다. 예비 측정에서 클래스명 변경 20%, id 변경 10%, 노드 삭제 3% 조건에서 여섯 방식 중 넷 이상이 60% 이상의 생존율을 보였다. 변화가 전혀 관측되지 않는 조건은 비교에 기여하지 않으므로 이보다 낮은 값을 채택하지 않았다.

셋째, 중간(Lv2)은 상한의 절반 내외로 두어 두 끝점 사이의 추이를 관측할 수 있도록 하였다.

따라서 Lv1과 Lv3는 각각 로케이터가 대체로 견디는 조건과 대응 단서가 소멸하는 조건을 뜻하며, 그 사이의 순서 관계만이 해석의 근거가 된다. 특정 수준이 실제 사이트의 어떤 변경 주기나 개편 규모에 대응한다고 해석해서는 안 된다.

실제 사이트의 시점별 DOM 변경 정도를 측정하여 파라미터를 보정하는 것이 바람직하나, 공개 아카이브의 스냅샷은 도메인 차단과 주차로 인해 동일 페이지의 연속 시점을 충분히 확보하기 어려웠다.

표 3. DOM 변형 유형별 세부 항목 및 스트레스 수준별 파라미터

Table 3. Detailed items and stress-level parameters of DOM transformation types

영역	세부 항목	Lv1	Lv2	Lv3
A. 속성 변형	A1. 클래스명 변경	접미사 추가, 20%	완전 치환, 50%	해시 난독화, 100%
	A2. id 변경·제거	10% 변경	50% 변경	100% 제거
	A3. 속성 순서 변경	적용	적용	적용
	A4. 무의미 속성 추가	노드당 1개	노드당 3개	노드당 5개
	A5. 클래스명 난독화	미적용	부분(50%)	전면, 8자 해시
B. 구조 변형	B1. 래퍼 노드 삽입	깊이 1, 10%	깊이 2, 35%	깊이 3, 60%
	B2. 태그 치환	10%	30%	60%
	B3. 형제 순서	인접 교환	블록 단위 이동	전역 재배치
	B4. 노드 깊이 변경	±1단계	±2단계	±3단계
	B5. 컨테이너 분할·병합	미적용	1회	2회
C. 노드 삽입삭제	C1. 광고 노드 삽입	1개	3개	5개
	C2. 오버레이 삽입	미적용	1개	2개
	C3. 장식 노드 삽입	노드당 0.1	노드당 1.0	노드당 2.0
	C4. 비필수 노드 삭제	3%	15%	30%

영역	세부 항목	Lv1	Lv2	Lv3
D. 텍스트·레이블 변형	D1. 레이블 문자열 변경	동의어 치환	표현 전면 교체	표현 전면 교체
	D2. 언어 전환	미적용	적용	적용
	D3. 텍스트→아이콘 대체	미적용	30%	70%
	D4. 공백·특수 문자 삽입	공백	공백+제로폭 문자	공백+제로폭 문자, 유사문자 80%
E. 복합 변형	E1. 다중 변형 결합	2중 결합	3중 결합	4중 이상

※ 각 수준의 수치는 저자가 설정한 상대적 스트레스 조건이며, 실제 사이트의 변경 정도를 측정한 값이 아니다.

이 문제는 5장에서 향후 과제로 다룬다.

강도 수준 간 비교에서는 동일 시드를 공유하는 대응표본 방식을 사용한다. 수준마다 다른 난수열을 사용하면 강도 차이와 난수에 의한 우연이 섞여, 강도를 높였음에도 생존율이 오히려 상승하는 구간이 나타난다. 또한 C4는 C1~C3이 삽입한 노드를 삭제 대상에서 제외한다. 그렇지 않으면 삭제가 삽입을 상쇄하여 강도가 역전된다.

3.5 비교 대상 지정 방식

본 논문에서 비교하는 요소 지정 방식은 표 4와 같이 여섯 가지이다. 웹 테스트 자동화와 데이터 수집에서 널리 사용되는 기본 방식으로 한정하였으며, 자가복구 로케이터는 재구현 비용을 고려하여 제외하였다.

L3와 L6는 모두 class를 사용하나 매칭 방법이 다르다. L3는 속성 전체를 완전일치로 비교하므로 토큰 하나만 바뀌어도 실패하는 반면, 실제 CSS 셀렉터(a, .a.b)와 셀레니움(Selenium)의

By.CLASS_NAME은 토큰 단위로 매칭한다. 완전 일치만 평가하면 class 기반 지정을 과소평가하게 되므로 두 방식을 모두 포함하였다.

표 4. 비교 대상 요소 지정 방식

Table 4. Element location strategies under comparison

구분	지정 방식	생성 방법
L1	절대 XPath	루트부터의 태그별 인덱스 경로
L2	ID 셀렉터	//*[@id="..."]
L3	class 셀렉터 (완전일치)	//tag[@class="..."]
L4	상대 XPath	최근접 식별 속성 조상 기준 상대 경로
L5	텍스트	normalize-space() 완전일치
L6	class 토큰	contains(concat(" ",@class," "), " tok ") 조합

지정 방식 k 의 생성 함수를 g_k , 로케이터 λ 를 문서 D 에서 해석한 결과 노드 집합을 $R(\lambda, D)$ 라 한다. 로케이터는 원본 문서에서 대상 노드 하나만을 지시할 때에만 유효하며, 식 (5)와 같이 정의한다.

$$g_k(D, t) = \begin{cases} \lambda, & R(\lambda, D) = \{t\} \\ \perp, & \text{otherwise} \end{cases} \quad (5)$$

복수 노드에 매칭되는 로케이터는 변형 이전에도 대상을 특정하지 못하므로 생성 실패로 처리한다. 이 조건을 두지 않으면 모호한 로케이터의 실패가 변형에 의한 실패로 잘못 집계된다.

4. 실험 및 결과

4.1 실험 환경

제한한 DOM 구조 변형 연산자를 이용하여, 변형 유형과 강도에 따른 요소 로케이터의 강인성을 측정하였다. 실험은 Google Colab 환경에서 Python 3.10과 lxml 라이브러리를 사용하여 구현하였다. 변형 연

산자는 시드 고정 난수를 사용하여 재현 가능하도록 구성하였으며, 변형 강도 간 비교에서는 동일 시드를 공유하는 대응표본 방식을 적용하였다. 실험은 변형 강도 조건별 생존율 비교, 변형 영역별 단독 적용, 탐색 대상별 비교의 세 가지로 구성하였다.

4.2 실험 데이터 구성

실험 대상 페이지는 각국 법원의 공개 차단 명령과 언론 보도에 명시된 불법 스트리밍 사이트 브랜드를 기준으로 수집하였다. 본 논문에서 브랜드란 도메인은 다르나 동일한 서비스명으로 운영되는 사이트군을 말한다. 차단명령에서도 도메인이 아닌 서비스명으로 대상을 특정하는 경우가 많다. 살아 있는 사이트에 직접 접속하지 않고 공개 아카이브(Internet Archive, Common Crawl)의 보존 스냅샷만을 이용하였다. 선행 연구[10]가 브라우저 자동화로 운영 중인 사이트에서 HTML을 직접 수집한 것과 달리, 아카이브를 사용하면 수집 시점과 무관하게 동일한 페이지를 재현할 수 있어 실험의 재현성이 확보된다. 다만 아카이브는 스크립트 실행 이전의 HTML을 보존하므로 동적으로 생성되는 요소는 관측할 수 없으며, 이는 4.6절에서 다룬다.

수집은 다음 순서로 수행하였다. 차단명령·보도에서 브랜드 26종을 추출하여 최상위 도메인을 조합한 도메인 330개 중, 서로 다른 2개 이상의 시점에 스냅샷이 존재하는 249개(75.5%)를 선별하였다. 이를 브랜드 단위로 통합한 뒤 브랜드별 대표 도메인에서 경로 패턴이 서로 다른 페이지를 최대 3개씩 수집하고, 구조 품질 검사와 태그 경로 집합 기반 중복 제거를 거쳐 13개 브랜드 22개 페이지를 확보하였다. 미통과 사유는 도메인 주차 페이지, 페이지 내용 소실, 링크 구조 빈약이었으며, 아카이브에 스냅샷이 존재한다는 사실만으로는 구조 분석에 사용할 수 없음을 보여준다. 브랜드 단위로 통합한 것은 동일 브랜드가 여

러 도메인으로 운영되는 특성상 도메인 수로 집계하면 표본이 부족해지기 때문이며, 실제로 fmovies는 6개, watchseries·putlocker·gomovies는 각 5개의 최상위 도메인에서 관측되었다.

확보한 22개 페이지 중 탐색 대상이 2개 미만인 1개를 제외하여 21개 페이지를 실험에 사용하였다. 초기 설계에서는 브랜드 단위로 dev와 test를 분리하였으나, 본 연구는 학습 모델을 포함하지 않고 변형 강도 파라미터만을 예비 보정하므로 전체 코퍼스를 사용하였다. 따라서 본 절의 결과는 관측된 조건에서의 비교 결과이며, 미관측 사이트에 대한 일반화 성능을 주장하지 않는다.

탐색 대상은 저자가 후보 목록을 검토하여 직접 지정하였다. 자동 탐지 휴리스틱을 예비 적용한 결과 페이지네이션을 콘텐츠 목록으로, 검색 모달 제목을 콘텐츠 제목으로 잡는 등의 오탐이 확인되었기 때문이다. 예비 실험에서 자동 탐지와 수동 지정의 텍스트 로케이터 생성률은 각각 32.3%와 18.5%로 크게 달랐다. 자동 탐지는 짧고 고유한 문자열을 제목으로 잡는 경향이 있어 텍스트 기반 지정 방식의 적용 가능성을 과대평가하며, 이는 요소 지정을 자동화할 경우 평가 결과가 낙관적으로 편향될 수 있음을 보여준다.

4.3 평가 방법

로케이터의 강인성은 생성률과 생존율로 평가한다. 실험은 원본 노드에 추적 속성을 부여하고 로케이터를 생성한 뒤, 변형을 적용하고 추적 속성을 제거한 사본에서 해당 로케이터를 해석하는 순서로 수행한다.

생성률은 전체 탐색 대상 중 해당 방식으로 유효한 로케이터를 만들 수 있는 대상의 비율로, 식 (6)과 같이 계산한다. 유효성의 정의는 3.5절의 식 (5)에 제시하였다.

$$\text{Cov}_k = \frac{|\{t \in T : g_k(D, t) \neq \perp\}|}{|T|} \times 100 \quad (6)$$

생존율(LSR)은 생성된 로케이터 중 변형 후에도 대상 노드를 정확히 지시하는 비율이다. 생성 가능한 대상의 집합을 $T_k = \{t \in T : g_k(D, t) \neq \perp\}$, 시드 집합을 S 라 할 때 식 (7)과 같이 정의한다. 여기서 지시 함수 $succ_k(t, \ell, s)$ 는 변형본 $D^{(\ell, s)}$ 에서 로케이터가 반환한 첫 번째 노드가 정답 노드 $\mu(t)$ 와 일치하면 1, 그렇지 않으면 0이다. 첫 번째 매칭만을 채점하는 것은 Selenium 등 실제 도구의 동작을 따른 것이다.

$$LSR_{k, \ell} = \frac{1}{|S \cap T_k|} \sum_{s \in S} \sum_{t \in T_k} succ_k(t, \ell, s) \times 100 \quad (7)$$

두 지표를 분리하여 집계하는 것이 중요하다. 합산하면 애초에 적용할 수 없는 방식과 변형에 취약한 방식이 구분되지 않기 때문이다.

본 실험에서는 변형 로그를 사용하지 않는다. 선행 연구[10]는 변형을 직접 생성하는 과정에서 기록한 변형 로그를 이용하여 정답 XPath를 갱신함으로써 성공률을 100%까지 회복시켰으나, 실제 추적 환경에서는 사이트 구조 변경에 대한 로그가 제공되지 않는다. 본 논문은 복구 기제가 없는 조건에서 원본에서 생성한 로케이터를 변형본에 그대로 적용하였을 때의 생존율을 측정한다. 추적 속성이 남아 있으면 로케이터가 이를 단서로 사용할 수 있으므로, 입력 문서에 추적 속성이 존재하지 않음을 매 시행마다 확인하였다.

4.4 변형 강도 조건별 로케이터 생존율

21개 페이지에서 지정한 탐색 대상 63개를 대상으로, 시드 10개와 여섯 가지 로케이터에 대해 총 15,120회 시행을 수행하였다. 불변조건 위반은 0건이었으며, 기준 조건(Lv0)에서 생성된 모든 로케이터가 정확히 대상을 지시함을 확인하였다 ($n=1,850$). 측정 결과는 표 5와 같으며, 강도 조건별 추이와 생성률은 그림 2에 나타내었다.

표 5. 스트레스 수준별 로케이터 생존율 및 유효 성공률 (%)
Table 5. Locator survival rate and effective success rate by stress level

조건	L1	L2	L3	L4	L5	L6
Lv0	100.0	100.0	100.0	100.0	100.0	100.0
Lv1	20.3	87.9	63.6	64.1	57.1	75.2
Lv2	1.3	48.6	21.2	32.9	7.6	28.8
Lv3	0.0	0.0	0.0	12.2	1.8	0.0
생성률	100.0	22.2	39.7	65.1	27.0	39.7
ESR(Lv1)	20.3	19.5	25.2	41.7	15.4	29.8
ESR(Lv2)	1.3	10.8	8.4	21.4	2.1	11.4
ESR(Lv3)	0.0	0.0	0.0	7.9	0.5	0.0

생존율은 로케이터를 생성할 수 있었던 대상만을 분모로 하므로, 생성률이 낮은 방식은 생존율이 높아도 실제 적용 범위가 좁다. 두 지표를 함께 고려하기 위해 유효 성공률(Effective Success Rate, ESR)을 식 (8)과 같이 정의한다. 유효 성공률은 전체 탐색 대상 중 해당 방식으로 지정할 수 있고 변형 후에도 살아남는 비율이다.

$$ESR_{k, \ell} = Cov_k \times LSR_{k, \ell} / 100 \quad (8)$$

절대 XPath는 생성률이 100%로 가장 높으나 Lv1에서 이미 20.3%로 급락하며, Lv2 이후에는 사실상 사용할 수 없다. 반면 ID 셀렉터는 Lv1에서 87.9%, Lv2에서 48.6%**로 완만하게 저하되지만 생성률이 22.2%에 그친다. 즉 탐색 대상의 5분의 1에 대해서만 적용할 수 있다. 실제 사이트에서 동일한 id 값이 문서 내에 중복 부여된 사례가 다수 확인되었으며, 이 경우 대상을 유일하게 지시할 수 없어 생성 실패로 처리된다. 검색 입력란의 id가 데스크톱용과 모바일용 요소에 동일하게 사용되는 경우가 여러 사이트에서 관측되었다.

class 지정 방식은 매칭 방법에 따라 결과가 달라진다. 완전일치(L3) 대비 토큰 방식(L6)의 생존율이 Lv1에서 11.6%p, Lv2에서 7.6%p 높았다. 생성률은 39.7%로 동일하였는데, 이는 class 문자열

전체가 유일하면 그 부분집합인 토큰 조합도 대체로 유일하기 때문이다. 실험에 사용한 탐색 대상 63개 중 19개(30.2%)가 둘 이상의 class 토큰을 가지고 있었다. 따라서 class 기반 지정을 완전일치로만 평가하면 실제 성능을 과소평가하게 된다.

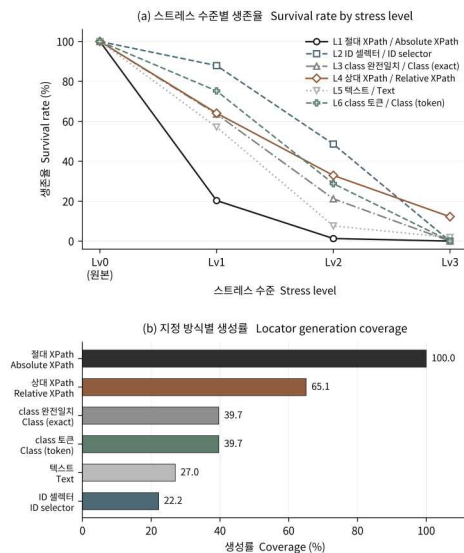


그림 2. 스트레스 수준별 생존율(a)과 지정 방식별 생성률(b)

Fig. 2. (a) Survival rate by stress level, (b) locator generation coverage

ESR 기준으로는 상대 XPath가 모든 조건에서 가장 높았다(Lv1 41.7%, Lv2 21.4%, Lv3 7.9%). 생성률이 65.1%로 두 번째로 높으면서 생존율도 중간 수준을 유지하기 때문이다. 다만 아래에서 보듯이 class 계열과의 차이는 통계적으로 유의하지 않다. 강인성과 적용 가능성이 상충하며, 생성률이 가장 높은 방식이 변형에 가장 취약하고 변형에 강한 방식은 적용 범위가 가장 좁다.

전체 시행을 합산한 백분율은 요소가 많은 페이지에 가중치가 실릴 수 있고, 동일 페이지에 여러 시드와 로케이터를 반복 적용하였으므로 개별 시행은 독립 표본이 아니다. 실제 독립 단위는 페이지이

므로 페이지별 ESR을 산출하여 분포와 검정 결과를 함께 제시한다. 페이지별 ESR은 해당 페이지의 전체 대상 중 성공한 비율로 정의하며, 생성 실패는 실패로 계산되므로 모든 페이지·로케이터 조합에 대해 값이 정의된다.

표 6. 페이지별 유효 성공률 분포(Lv2, n=21)

Table 6. Per-page distribution of effective success rate at stress level 2 (n=21)

로케이터	중앙값 (%)	사분위 범위(IQR)	평균±표준편차
L1	0.0	[0.0, 0.0]	1.3 ± 2.4
L2	0.0	[0.0, 20.0]	10.3 ± 13.7
L3	3.3	[0.0, 10.0]	7.8 ± 10.6
L4	25.0	[5.0, 33.3]	21.4 ± 16.9
L5	0.0	[0.0, 0.0]	1.7 ± 4.5
L6	7.5	[0.0, 20.0]	11.5 ± 11.2

페이지별 유효 성공률 분포는 표 6과 같으며, 그림 3에 나타내었다. 상자는 사분위 범위, 가로선은 중앙값, 점과 막대는 평균과 표준편차를 나타낸다.

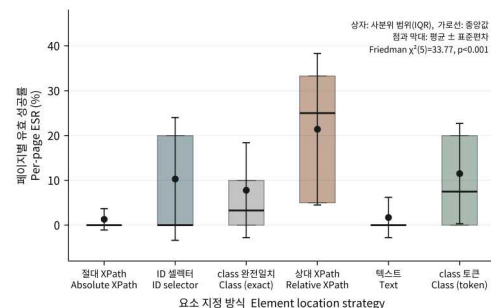


그림 3. 페이지별 유효 성공률 분포(Lv2, n=21)

Fig. 3. Per-page distribution of effective success rate at Lv2 (n=21)

프리드만 검정(Friedman test) 결과 로케이터 간 차이는 Lv2($\chi^2(5)=33.77, p<0.001$)와 Lv1($\chi^2(5)=15.34, p=0.009$) 조건에서 모두 유의하였다. Lv3는 모든 로케이터의 중앙값이 0이어서 해석이 어려우므로 보고하지 않는다.

사후 비교는 윌콕슨 부호순위검정(Wilcoxon signed-rank test)에 홀름 보정(Holm correction)을 적용하였다. 이하 보고하는 p는 모두 보정 후 값이다. Lv2에서 유의한 차이가 확인된 쌍은 상대 XPath와 절대 XPath(쌍별 차이의 중앙값 +25.0%p, $p=0.006$), 상대 XPath와 텍스트(+17.5%p, $p=0.009$), class 토큰과 절대 XPath(+6.7%p, $p=0.010$), class 토큰과 텍스트(+7.5%p, $p=0.039$), 상대 XPath와 ID 셀렉터(+2.5%p, $p=0.039$)이다.

상대 XPath와 class 계열의 차이는 유의하지 않았으며(class 토큰 $p=0.304$, class 완전일치 $p=0.106$), 완전일치와 토큰 방식의 차이도 유의 수준에 도달하지 못하였다($p=0.068$). 페이지별 편차가 크고 표본이 21개에 그치기 때문이다. 따라서 ESR 순위는 참고 지표로 제시하며, 통계적으로 확인된 것은 절대 XPath와 텍스트 로케이터가 상대 XPath 및 class 토큰 방식보다 유의하게 낫다는 점이다.

4.5 변형 영역별 실패 분포

각 변형 영역을 단독으로 적용하여 어떤 영역이 어떤 로케이터를 무력화하는지 분석하였다. 노드 삽입과 삭제는 인덱스 이동을 서로 상쇄할 수 있으므로 C-i(삽입)와 C-d(삭제)로 분리하여 관측하였다. 강도를 높일수록 생존율이 낮아지는 관계가 모든 조건에서 유지되었으며, 불변조건 위반도 0건이었다. 영역별 측정 결과는 그림 4와 같다.

각 영역이 서로 다른 로케이터를 선택적으로 무력화하는 대각 패턴이 뚜렷하게 나타난다. A 속성 변형은 속성값에 의존하는 L2·L3·L4·L6를 무력화하며, 경로 기반 L1과 텍스트 기반 L5에는 전혀 영향을 주지 않는다. B 구조 변형은 L1을 완전히 무력화하고(17→1→0) L4를 점진적으로 저하시킨다(70→57→50). L5도 100→97%로 미세하게 저하되는데, 형제 노드 순서가 바뀌면 상위

컨테이너의 텍스트 연결 순서가 달라져 완전일치 조건이 어긋나기 때문이다. C 노드 삽입·삭제는 L1에 집중적으로 영향을 주어 삽입은 51%, 삭제는 59%까지 저하시킨다. 다른 로케이터가 거의 영향을 받지 않는 것은 경로 표현이 태그별 인덱스를 사용하므로 서로 다른 태그의 노드가 추가·제거되어도 경로가 유지되기 때문이다. D 텍스트 변형은 L5만을 선택적으로 무력화한다(59→10→0).

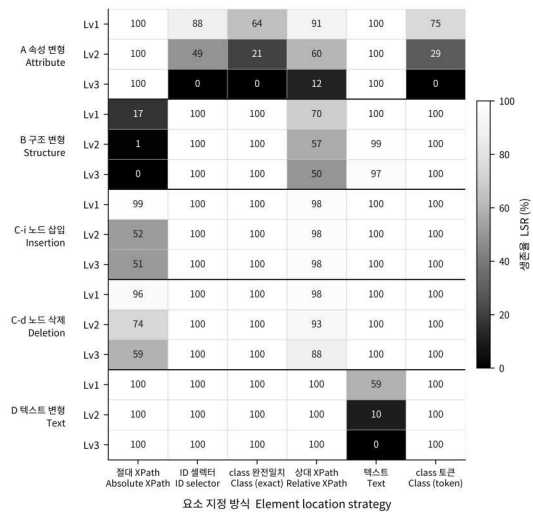


그림 4. 변형 영역 단독 적용 시 로케이터 생존율
Fig. 4. Locator survival rate under single-area transformation

완전일치(L3)와 토큰 방식(L6)의 차이는 전적으로 A 영역에서 발생한다. B·C·D 영역에서는 두 방식이 모두 100%로 동일하며, A 영역에서만 Lv1 64% 대 75%, Lv2 21% 대 29%로 토큰 방식이 우세하다. 속성 변형이 class 토큰의 일부만 바꿀 때 완전일치는 실패하지만 토큰 방식은 남은 토큰으로 대상을 계속 지시할 수 있기 때문이다.

선행 연구[10]의 변형 규칙은 노드 삽입만을 포함하고 삭제를 다루지 않는다. 삭제 단독 적용 시 절대 XPath가 59%까지 저하된 결과는 연산자 확장을 통해 새로 관측된 것이다.

4.6 탐색 대상별 결과

탐색 대상별 결과는 표 7과 같다. T2(콘텐츠 항목 목록)의 생성률이 43.8%로 가장 낮다. 목록 컨테이너는 일반적으로 id가 없고 클래스명이 레이아웃 목적으로 부여되어 다른 컨테이너와 중복되는 경우가 많다. T5(다음 항목 링크)는 생성률이 53.3%로 중간 수준이나 스트레스 수준2(Lv2) 생존율이 5.6%로 가장 낮다.

표 7 탐색 대상별 생성률 및 Lv2 생존율
Table 7. Coverage and survival rate at Lv2 by target element

대상	페이지	생성률(%)	Lv2 생존율(%)
T2	16	43.8	13.4
T3	18	55.6	11.2
T4	21	49.5	27.1
T5	6	53.3	5.6

※ 생성률은 여섯 가지 지정 방식 전체에 대한 평균이다.

텍스트가 짧고 클래스명이 단순하며 얇은 컨테이너에 위치하여 어떤 방식으로든 안정적으로 지시하기 어렵다. T4(검색 입력)는 모든 페이지에 존재하나 생성률은 49.5%에 그친다. 이는 name 속성이 부여되는 경우가 많음에도 앞서 언급한 id 중복 문제로 ID 선택터를 생성할 수 없는 사례가 다수 발생하였기 때문이다.

T1(플레이어 컨테이너)은 21개 페이지 중 2개에서만 관측되어(생성률 70.0%, Lv2 생존율 28.6%) 표 7의 탐색 대상별 집계에서 제외하고 탐색적 관찰로만 기술한다. 브랜드 13종에서 재생 경로로 판단되는 페이지를 수집하였으나 정적 HTML에 플레이어 노드가 존재한 것은 2개 페이지에 그쳤으며, URL이 /episodes/watch-... 형태인 페이지에서도 <video> 및 <iframe> 요소가 발견되지 않는 사례가 다수였다. 이는 플레이어가 JavaScript에 의해 런타임에 주입되기 때문으로 판단된다. 아카이브는 스크립트 실행 이전

의 HTML을 보존하므로, 공개 아카이브의 정적 스냅샷을 이용하는 한 플레이어 컨테이너는 안정적인 관측 대상이 될 수 없다. 합성 페이지나 오픈소스 플레이어 템플릿으로 표본을 보강하는 방안도 가능하나, 실제 사이트가 아닌 인공 페이지로 추정하는 것이 되어 본 실험의 전체와 상충하므로 채택하지 않았다.

이와 관련하여 본 논문의 실측 대상은 콘텐츠 항목 목록, 제목, 검색 입력, 다음 항목 링크에 집중되어 있다. 이들은 불법 스트리밍 사이트에 특유한 요소라기보다 콘텐츠 제공 웹사이트 전반에 공통되는 요소이다. 따라서 본 논문의 결과는 불법 스트리밍 사이트에서 수집한 실제 DOM을 대상으로 하되, 그 함의는 구조가 빈번히 변경되는 웹사이트의 일반적 요소 추적에 가깝게 해석하는 것이 정확하다.

5. 결론

본 논문에서는 불법 스트리밍 사이트의 화면 구조가 차단 회피 과정에서 반복적으로 변경됨에 따라 기존 요소 로케이터가 무력화되는 문제를 다루었다. 선행 연구[10]는 변형 로그를 이용하여 정답 로케이터를 복구하는 방법을 제시하였으나, 실제 추적 환경에서는 변형 로그가 제공되지 않는다. 본 논문은 복구 기제가 없는 조건에서 여섯 가지 요소 지정 방식의 생성률과 생존율을 비교 추정하였다. 이를 위해 선행 연구의 변형 규칙에 노트 삭제와 텍스트 변형을 추가하고 세 단계의 강도 조건을 도입하여 변형 연산자를 다섯 개 영역으로 재구성하였으며, 공개 아카이브에서 확보한 13개 브랜드 21개 페이지를 대상으로 15,120회의 시행을 수행하였다.

실험 결과, 강인성과 적용 가능성이 상충하는 것으로 나타났다. 절대 XPath는 모든 대상에 적용할 수 있으나 스트레스 수준1(Lv1)에서 이미 생존율이 20.3%로 급락하였고, ID 선택터는 변형에 상대적으

로 강인하지만 생성률이 22.2%에 그쳤다. 두 지표를 결합한 유효 성공률에서는 상대 XPath가 Lv2 21.4%로 가장 높았으나, 페이지 단위 통계 검증에서 class 계열과의 차이는 유의하지 않았다. 생성률을 제약하는 주요 요인은 실제 사이트에서 빈번하게 관측된 id 속성의 중복 부여였다. 이는 검색 입력란의 식별자가 데스크톱용과 모바일용 요소에 동일하게 부여되는 등 실제 사이트에서 흔히 나타나는 현상으로, 요소 지정을 수작업으로 수행하는 경우에는 드러나지 않으므로 자동화 파이프라인 설계 시 고려되어야 한다.

또한 변형 영역별로 무력화되는 지정 방식이 명확히 구분되었다. 속성 변형은 속성 기반 로케이터만, 구조 변형은 경로 기반 로케이터만, 텍스트 변형은 텍스트 기반 로케이터만 선택적으로 무력화하였으며, 이는 여러 방식을 함께 사용하는 접근이 유효할 수 있음을 뜻한다. 한편 class 기반 지정을 완전 일치 대신 토큰 방식으로 구현하면 생존율이 높아지고, 탐색 대상을 자동으로 지정하면 텍스트 로케이터 생성률이 과대평가되어, 구현 방식과 대상 지정 방식이 평가 결과에 직접 영향을 미침을 보였다.

본 연구의 결과는 침해 대응 수집 파이프라인의 설계에 다음과 같이 적용할 수 있다.

첫째, 지정 방식을 하나로 고정하지 않고 복수로 생성하여 함께 보관한다. 여섯 방식 중 하나라도 성공한 비율은 Lv2에서 32.4%로, 단독 최고인 상대 XPath(21.4%)보다 11.0%p 높았다. 수집 시점에 대상마다 절대 XPath, 상대 XPath, class 토큰을 함께 생성해 두고 재방문 시 순차적으로 시도하면 추적 지속성이 향상되며, 구현 비용은 로케이터 문자열을 추가로 저장하는 정도에 그친다.

둘째, class 기반 지정은 완전일치 대신 토큰 방식으로 구현한다. 두 방식은 생성률이 39.7%로 동일하나, 생존율은 토큰 방식이 Lv1에서 11.6%p, 수준 2에서 7.6%p 높았다. 구현 방식만 바꾸어 얻을 수 있는 개선이므로 우선 적용할 수 있다.

셋째, 탐색 대상 지정을 자동화할 경우 그 결과를 검증하는 절차가 필요하다. 자동 탐지와 수동 지정의 텍스트 로케이터 생성률이 각각 32.3%와 18.5%로 크게 달랐으며, 이는 자동 탐지가 짧고 고유한 문자열을 제목으로 잡는 경향에서 비롯된다. 대상 지정이 잘못되어도 로케이터 자체는 정상 동작하므로 파이프라인은 정상으로 보이거나 수집 결과가 어긋난다. 따라서 자동 지정 결과에 대해서는 표본 검사나 기능 검증을 병행할 필요가 있다.

넷째, id 속성의 유일성을 전제하지 않아야 한다. 실험에 사용한 페이지에서 검색 입력란의 id가 데스크톱용과 모바일용 요소에 중복 부여된 사례가 다수 관측되었다. ID 선택터의 생성률이 22.2%에 그친 주요 원인이 이것이며, 수집기가 첫 번째 매칭 요소를 반환하는 방식으로 동작하면 의도하지 않은 요소를 지정하게 된다.

이러한 개선은 저작권 보호 기관과 침해 대응 사업자의 수집 시스템에 직접 적용할 수 있다. 특히 복수 로케이터 보관과 class 토큰 방식 채택은 기존 파이프라인의 구조를 바꾸지 않고 적용할 수 있으므로 도입 비용이 낮다.

다만 본 연구는 다음의 한계를 가진다. 스트레스 수준이 저자가 설계한 인공적 구분이라는 점, 실험 대상이 21개 페이지로 제한적이어서 상위 방식 간 차이를 통계적으로 구분하지 못한 점, 그리고 기존 강인화 기법과의 비교를 포함하지 않은 점이다.

이에 따라 향후 연구는 세 방향으로 진행할 수 있다. 첫째는 스트레스 수준의 실측 보정이다. 동일 사이트의 시점별 스냅샷 쌍으로 인공 변형과 실제 구조 변화의 결과가 유사한지 검증하고 파라미터를 조정한다. 다만 공개 아카이브는 도메인 차단과 주차로 연속 시점의 확보가 어려우므로 대상 도메인을 직접 주기적으로 수집하는 관측 체계가 필요하다. 둘째는 변형 유형에 따른

지정 방식의 동적 선택이다. 본 논문에서 확인된 영역별 상보성을 활용하여 두 시점의 문서를 비교해 변경 영역을 추정하고 그에 맞는 방식을 선택하는 기법을 설계할 수 있으며, 이때 추정 정확도와 선택으로 얻는 이득의 상한을 함께 규명해야 한다. 셋째는 강인화 기법과의 비교다. ROBULA+[11], SIDEREAL[12], Similo[14]는 다수의 단서를 종합하거나 정제 변환을 반복하므로 본 논문의 변형 조건에서 다른 양상을 보일 수 있으며, 동일한 변형 코퍼스에 적용하여 비교할 필요가 있다. 이와 함께 렌더링 이후의 DOM을 확보할 수 있는 환경에서 플레이어 영역을 포함한 측정도 과제로 남는다.

본 연구는 문화체육관광부 및 한국문화기술기획평가원의 재원으로 2026년도 SW저작권기술(+법) 융합인재양성사업의 지원을 받아 수행되었음(과제명 : OTT 콘텐츠 저작권보호기술개발 및 적용을 위한 저작권기술(+법)융합인재양성, 과제번호 : RS-2023-00225267, 기여율: 100%)

참 고 문 헌

- [1] Korea Copyright Protection Agency, Annual Report on Copyright Protection 2026 (Based on 2025 Survey), Korea Copyright Protection Agency, Feb. 2026. <https://www.kcopa.or.kr/>
- [2] Korea Copyright Protection Agency, 2024 Survey on Overseas Infringement of Korean Wave Content: Focusing on Illegal Distribution Sites in Film, Broadcasting, and Webtoon Sectors, KP Research 2024-02, Dec. 2024. <https://nsp.nanet.go.kr/>
- [3] Korea Copyright Protection Agency, 2023 Survey on Overseas Infringement of Korean Wave Content: Focusing on Illegal Distribution Sites in Film, Broadcasting, and Webtoon Sectors, KP Research 2023-01, Nov. 2023. <https://www.kcopa.or.kr/>
- [4] Ministry of Culture, Sports and Tourism, First Emergency Blocking Order Issued Against 34 Copyright-Infringing Sites, Press Release, May 11, 2026. <https://www.mcst.go.kr/>
- [5] B.-C. Park and Y.-M. Kim, "A Study on Video Transformation Type Analysis and Robustness Evaluation Criteria for Illegal Streaming Environments," *Journal of Software Forensics*, vol. 22, no. 2, pp. 79-91, 2026. DOI: 10.29056/jsf.2026.06.08
- [6] B.-C. Park and Y.-M. Kim, "An Input Video Normalization Method Based on Valid Regions and Frames for Illegal Streaming Video Identification," *Journal of Software Forensics*, vol. 22, no. 2, pp. 93-105, 2026. DOI: 10.29056/jsf.2026.06.09
- [7] I. Yoo, J.-C. Lee, S. Jang, B.-C. Park, S.-Y. Kim, and Y.-M. Kim, "An Illegal Distribution Platform Tracking Method Using IP Port Scanning," *Journal of Software Forensics*, vol. 19, no. 4, pp. 115-122, Dec. 2023. DOI: 10.29056/jsav.2023.12.12
- [8] S.-W. Oh, M.-C. Park, and Y.-M. Kim, "Utilizing Advertiser Unique Identification Numbers for Illegal Copyright Infringement Site Feature Information Collection Strategy," *Journal of Software Forensics*, vol. 20, no. 3, pp. 43-52, Sep. 2024. DOI: 10.29056/jsav.2024.09.05
- [9] S.-H. Park, I.-J. Yoo, S.-Y. Jang, B.-C. Park, S.-H. Shin, S.-Y. Kim, and Y.-M. Kim, "An Illegal Streaming Site Structure Analysis Method Based on Text Numericalization," *Journal of Software Forensics*, vol. 20, no. 4, pp. 193-201, Dec. 2024. DOI: 10.29056/jsav.2024.12.20
- [10] J. Lee, B. Park, U. Jang, and Y. Shin, "A Function-Centric Anchor-Based HTML Augmentation and an Adaptive XPath Framework for Tracking Illegal OTT Streaming Sites," in *Human-Computer Interaction (HCII 2026)*, LNCS vol. 16701, Springer, pp. 490-509, 2026. DOI:

- 10.1007/978-3-032-29583-5_30
- [11] M. Leotta, A. Stocco, F. Ricca, and P. Tonella, "ROBULA+: An Algorithm for Generating Robust XPath Locators for Web Testing," *Journal of Software: Evolution and Process*, vol. 28, no. 3, pp. 177-204, 2016. DOI: 10.1002/smr.1771
- [12] M. Leotta, F. Ricca, and P. Tonella, "SIDEREAL: Statistical Adaptive Generation of Robust Locators for Web Testing," *Software Testing, Verification and Reliability*, vol. 31, 2021. DOI: 10.1002/stvr.1767
- [13] V. Nguyen, T. To, and G.-H. Diep, "Generating and Selecting Resilient and Maintainable Locators for Web Automated Testing," *Software Testing, Verification and Reliability*, vol. 31, no. 3, e1760, 2021. DOI: 10.1002/stvr.1760
- [14] M. Nass, E. Alégroth, R. Feldt, M. Leotta, and F. Ricca, "Similarity-Based Web Element Localization for Robust Test Automation," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 3, Article 75, Apr. 2023. DOI: 10.1145/3571855
- [15] R. Coppola, R. Feldt, M. Nass, and E. Alégroth, "Ranking Approaches for Similarity-based Web Element Location," *Journal of Systems and Software*, vol. 222, Article 112286, 2025. DOI: 10.1016/j.jss.2024.112286
- [16] M. Nass, E. Alégroth, and R. Feldt, "Improving Web Element Localization by Using a Large Language Model," *Software Testing, Verification and Reliability*, vol. 34, e1893, 2024. DOI: 10.1002/stvr.1893
- [17] M. Nass, E. Alégroth, and R. Feldt, "Why Many Challenges with GUI Test Automation (Will) Remain," *Information and Software Technology*, vol. 138, 106625, 2021. DOI: 10.1016/j.infsof.2021.106625
- [18] H. Kirinuki, H. Tanno, and K. Natsukawa, "COLOR: Correct Locator Recommender for Broken Test Scripts using Various Clues in Web Application," *Proc. 26th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 310-320, 2019. DOI: 10.1109/SANER.2019.8667976
- [19] M. Nass, E. Alégroth, R. Feldt, and R. Coppola, "Robust Web Element Identification for Evolving Applications by Considering Visual Overlaps," *Proc. IEEE Conference on Software Testing, Verification and Validation (ICST)*, pp. 258-268, 2023.
- [20] S. Di Meglio, L. L. L. Starace, V. Pontillo, R. Opdebeeck, C. De Roover, and S. Di Martino, "E2EGit: A Dataset of End-to-End Web Tests in Open Source Projects," *Proc. IEEE/ACM 22nd International Conference on Mining Software Repositories (MSR)*, pp. 836-840, 2025.

저 자 소 개



장의진(Uijin Jang)

1999.8 숭실대학교 컴퓨터학부 졸업
 2002.8 숭실대학교 컴퓨터학과 석사
 2011.2 숭실대학교 컴퓨터학과 박사
 2018.9-현재 : 숭실대학교 스파르탄AI교육원 부교수
 <주관심분야> 인공지능, 네트워크 및 정보 보호, 디지털 저작권 보호 기술



이지은(Jieun Lee)

2023.2 경운대학교 항공정보통신학과 졸업
2023.3-현재 : 숭실대학교 컴퓨터학과 석박
통합과정
<주관심분야> 컴퓨터네트워크, 인공지능,
클라우드, 정보보안



임형민(Hyungmin Lim)

2001.2 숭실대학교 컴퓨터공학과 졸업
2003.2 숭실대학교 컴퓨터학과 석사
2011.2 숭실대학교 컴퓨터학과 박사
2015.6-현재 : (주)엘엔제이테크 대표이사
<주관심분야> 정보보안, 인공지능보안, 보안평가