https://doi.org/10.7840/kics.2025.50.7.1118

다크웹 프로파일링을 위한 캡챠 분류 자동화 프레임워크

유 은 선•, 박 규 나*, 백 서 이*, 김 성 민°

CAPTCHA Classification Framework for Dark Web Profiling

Eunseon Yu•, Gyuna Park*, Seo-Yi Baik*, Seongmin Kim°

요 약

최근 익명성을 보장하는 토르 네트워크 기반 다크웹 내에서의 범법 행위가 기하급수적으로 급증하고 있다. 불법적인 거래가 증가함에 따라, 다크웹 생태계에 존재하는 암시장의 형태를 파악하는 것이 수사에 있어서 중요하다. 그러나 다크웹 운영자들은 정보의 누설을 최소화하기 위해 자동화된 크롤링 봇을 제한한다. 그 중 캡챠(CAPTCHA)는 범용적으로 사용되며 다양한 형태로 존재한다. 이에 본 논문은 캡챠를 활용하여 크롤링 봇의 접근을 차단하거나 제한하는 암시장의 형태를 파악하는 자동화 프레임워크를 제시한다. 다크웹 상에서 존재하는 캡챠의 분류 기준을 제시한 뒤, 분류를 자동화할 수 있는 모델을 제시한다. 캡챠 분류 모델은 다크웹 상의 은어를 기반으로 한 크롤링 통해 120여 개의 링크를 수집한 뒤, 캡챠 분류 기준에 따라 유형에 맞추어 분류한다. 제안한 프레임워크는 93.33%의 준수한 다크웹 내 캡챠 분류 정확도 성능을 보였으며, 이를 기반으로 캡챠를 우회하는 자동화된 크롤링 봇을 통해 다크웹 프로파일링의 효율성을 증대시킬 것으로 기대한다.

Key Words : Dark Web, Crawling Bot, Bot Detection, CAPTCHA

ABSTRACT

Recently, criminal activities on the dark web, facilitated by anonymity through Tor, have exponentially increased. With the rise in illegal transactions, understanding the structure of black markets within the dark web ecosystem has become crucial for investigations. However, dark web operators actively minimize information leakage by restricting automated crawling bots. Among these measures, CAPTCHA is widely utilized and exists in various forms. This paper proposes an automated framework to identify and analyze the structure of black markets that use CAPTCHA to block or restrict access by crawling bots. It first introduces classification criteria for CAPTCHAs on the dark web and then presents a model capable of automating their classification. The CAPTCHA classification model collects over 120 links through crawling based on the jargon and terminology used in the dark web, then categorizes them according to the established CAPTCHA classification criteria. The proposed framework demonstrated a commendable CAPTCHA classification accuracy of 93.33% within the dark web. Based on this, it is expected to enhance the efficiency of dark web profiling through an automated crawling bot capable of bypassing CAPTCHA mechanisms.

※ 본 연구는 2024년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. RS-2024-00351898)과 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0 사업의 연구결과로 수행되었음 (No. IITP-2022-RS-2022-00156310)
◆ First Author : Sungshin University Department of Convergence Security Engineering, 54ryues@gmail.com, 학생회원
° Corresponding Author : Sungshin University Department of Convergence Security Engineering, sm.kim@sungshin.ac.kr, 정회원
* Sungshin University Department of Convergence Security Engineering, 20211099@sungshin.ac.kr; 0801baik@gmail.com, 학생회원
논문번호 : 202501-007-D-RU, Received January 3, 2025; Revised February 9, 2025; Accepted February 11, 2025

Ⅰ. 서 론

최근 전 세계적으로 다크웹(Dark Web)을 통한 불법적인 행위가 기하급수적으로 늘어나고 있으며, 이를 기반으로 거래가 이루어지는 암시장(Black Market)의 규모 또한 커지고 있다 [1]. 구체적으로, 다크웹에서 주로 행해지는 범죄는 마약 거래와 여권, 인증서를 위조해 주는 서비스를 넘어 악성 코드를 사고 팔거나 랜섬웨어 서비스를 제공하는 등의 다양한 범법 행위를 포함한다.

이러한 암시장 생태계는 다크웹의 핵심 기반 기술인 토르 네트워크(Tor network)가 보장하는 익명성에 근간을 둔다. 표면 웹(Surface Web)과 달리 토르 히든 서비스(Tor Hidden service)를 통해 판매자는 운용하는 서버의 익명성을 보장받을 수 있고, 구매자 또한 토르 브라우저를 통해 다크웹에 접속하기 때문에 다크웹 내 트래픽은 발신자 익명성과 수신자 익명성을 보장한다 [2]. 이러한 다크웹 속성으로 인해, 다크웹을 대상으로 한 수사에는 많은 어려움이 존재한다.

다크웹 상에서 이루어지는 범죄에 대한 수사를 위해서는, 다크웹 생태계의 전반적인 이해와 운영자를 프로파일링하기 위한 다양한 증거들을 수집하는 것이 필수적이다. 이를 위해, 선행 연구에서는 다양한 다크웹 내 데이터 수집을 위해 크롤링 기반의 다크웹 분석 프레임워크들을 제안하였다 [3]. 대규모 데이터 수집을 위해서는 이러한 자동화된 형태의 크롤러 사용이 불가피하며, 수집한 HTML 태그들을 바탕으로 비트코인 지갑주소, 표면 웹에서 사용하는 계정 정보와의 매칭 등 수사기관에서 활용할 수 있는 증거들을 확보할 수 있다.

그러나 최근 다크웹 운영자들 또한 이에 대한 인지를 바탕으로, 자동화된 크롤링 봇의 접근을 통제하기 위해 봇으로 추측되는 트래픽을 차단하거나 캡챠(CAPTCHA)를 적극적으로 활용하고 있다 [4]. 이러한 캡챠의 사용은 다크웹으로부터의 정보 수집을 어렵게 하며, 나아가 수사를 위한 증거 확보를 제한한다. 특히, 다크웹 사이트 운영자 중 일부는 표면 웹에서도 사용되고 캡챠 우회 방법이 밝혀진 범용 캡챠를 이용하는 것이 아닌, 자체 제작한 커스터마이징(Customizing) 캡챠를 사용하는 경우가 다수 존재함을 본 연구를 통해 식별하였다. 따라서, 원활한 수사 및 데이터 수집을 위해서는 다크웹 내 존재하는 다양한 캡챠의 형태를 파악하고, 각 기법에 대한 우회 방법을 도출하여야 한다.

본 논문에서는 다크웹 랜딩 페이지 내 HTML 분석을 기반으로, 다크웹 운영자가 크롤러로 인한 정보 유출을 방지하기 위해 활용하는 주요 수단인 캡챠의 분류 기준을 제시하며 분류를 자동화할 수 있는 프레임워크를 제안한다. 이를 위해, 실제 다크웹 내 불법 거래를 위해 운영되는 onion 사이트들을 수집하고 각 사이트에 적용된 캡챠 기법에 대한 분석을 진행하였다. 본 연구를 통해서 수사기관에서 다크웹 생태계 분석을 위해 일차적으로 우회해야 할 크롤링 봇 탐지 기법을 식별하고, 각 기법별로 캡챠를 우회하는 방안을 제시함으로써, 다크웹 수사에 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 배경지식을 다루며, 3장에서는 관련 연구를 검토하고 연구 목표를 제시한다. 4장에서는 다크웹의 크롤링 봇 탐지/차단 기술을 설명하고 표본을 바탕으로 캡챠를 수집한다. 5장에서는 수집 과정에서 사용된 자동화 프레임워크를 소개하며, 6장에서는 실험 결과를 분석한다. 이어서 7장에서는 연구의 한계점인 논의 및 고찰을 다루고, 8장에서 연구 결과를 종합적으로 정리한다.

Ⅱ. 배경지식

2.1 토르 네트워크와 다크웹

토르 네트워크는 3-hop onion 라우팅을 기반으로 트래픽을 암호화하고, 여러 국가에 분포된 토르 노드들을 경유함으로써 높은 익명성을 보여준다. 토르는 '히든 서비스'를 제공하는데, 웹 서비스를 제공하는 서버의 IP 주소를 감출 수 있도록 하는 기술이다[5]. 다시 말해, 토르 히든 서비스를 이용하면 웹 사이트 제공자도 토르 브라우저를 이용하는 클라이언트와 같이 자신의 IP 주소를 노출하지 않고 웹 사이트를 호스팅 할 수 있다. 이러한 서비스는 일반 웹 사이트와 달리 '해시암호.onion' 형식의 도메인으로 끝나는 주소를 입력해야 하고 일반 웹 브라우저로는 접근할 수 없으며, 토르 브라우저를 통해서만 접근할 수 있다.

다크웹은 토르 네트워크를 기반하여, 토르 onion 사이트 운영자가 판매하는 불법적인 물품을 토르 이용자가 토르 브라우저를 통해 구매하는 생태계를 의미한다. 이처럼 다크웹 내에서는 sender/receiver 익명성이 모두 보장되기 때문에 이를 기반으로 한 불법적인 행위에 대한 수사에 많은 어려움이 존재한다. 따라서, 수사기관에서는 시드(seed)가 될 수 있는 onion 사이트들의 정보를 꾸준히 수집, 모니터링 및 분석하여 범죄자 프로파일링을 위한 수사에 활용하기 위해 노력하고 있다.

다크웹 내에서 onion 사이트들을 수집하기 위한 방안으로, 딥/다크웹 검색 엔진 또는 아카이빙(archiving) 사이트를 활용할 수 있다. 아카이빙 사이트는 검색 결과에 따른 관련 히든 서비스의 가동 시간과 상태를 지속적으로 확인하여 키워드별로 관련성 높고 액세스 가능한

페이지를 표시한다. 해당 사이트의 아카이빙된 링크들은 누구나 무료로 사용할 수 있으며 리스팅 또한 허용한다[6]. 대표적인 무료 아카이빙 사이트로는 비영리 커뮤니티 프로젝트로 FOSS 개발자가 운영하는 'Tor Link'가 있으며, 본 연구에서는 'Tor Link' 아카이빙 사이트를 활용하여 다크웹 링크를 수집하는 크롤러를 제작하였다.

2.2 캡챠(CAPTCHA)

캡챠는 웹 사이트에서 기계는 인식할 수 없으나 사람은 쉽게 인식할 수 있는 텍스트 및 이미지를 통해 사람과 기계를 구별하는데 사용되는 튜링 테스트이다[7]. 봇은 이 테스트를 통과할 가능성이 작으므로 캡챠를 풀지 못하고, 웹 서버 또는 애플리케이션과의 상호작용이 차단된다. 일반적으로 캡챠는 스팸 봇이 포럼에 자동으로 포스팅하는 것을 방지하거나, 크롤링 봇으로부터의 웹 사이트 데이터 보호를 목적으로 사용된다[8].

캡챠에는 텍스트 및 이미지 기반 캡챠, 논리 퍼즐 캡챠, 체크 박스 캡챠 등 다양한 종류가 있다. 텍스트 기반 캡챠는 사용자에게 왜곡된 텍스트를 보여주고 이를 올바르게 입력하도록 요구한다. 이미지 기반 캡챠는 특정한 이미지를 선택하거나, 이미지 속의 객체를 식별하도록 요구한다. 과거에는 이와 같은 텍스트 및 이미지 기반 캡챠의 사용 빈도가 가장 높았다. 그러나 최근에는 캡챠 기술이 진화하여 문제를 풀거나 퍼즐을 맞추도록 요구하는 논리 퍼즐 캡챠, '로봇이 아닙니다. (I'm not a robot.)'이라는 문구 옆의 체크 박스를 클릭하면 되는 체크 박스 캡챠가 자주 사용된다. 체크 박스 캡챠는 사람이 체크 박스를 클릭하기 전까지의 모든 일련의 과정을 확인하여 컴퓨터와 사람의 행위는 확연히 구별된다.

단순한 형태의 캡챠를 통해서도 봇을 차단할 수 있었던 과거와 달리, 기계 학습과 같은 인공지능 기술의 발전에 따라 고도화된 봇은 왜곡된 캡챠 이미지나 텍스트를 식별할 수 있게 되었다. 이로 인해 더 난해한 노이즈나 이미지를 추가한다거나 시간 제한을 두는 등, 캡챠가 더욱 복잡한 형태로 진화하고 있다. 실제로, 다크웹에서 사용되는 캡챠 중 일부는 표면 웹에서 일반적으로 나타나는 형태의 캡챠와 달리 커스터마이징되어 있음을 확인할 수 있다. 따라서 자동화된 다크웹 내 데이터 및 증거 수집을 위해서는 적용된 캡챠 기법과 이를 우회할 방법에 관한 연구가 선행되어야 한다.

2.3 표면 웹과 다크웹의 캡챠 비교

일반적인 표면 웹 내에서와 다크웹 상에서의 서버 운영자는 캡챠 사용의 목적이 다르다. 표면 웹은 사용자 인증 및 분산 서비스 거부 공격 차단이 주요 목적이며, 다크웹은 정보 수집을 위한 스크래핑을 막는 것이 목적이다. 구체적으로, 표면 웹에서는 일반적으로 사용자의 편의와 보안의 균형을 맞춘 캡챠가 주로 사용된다. 사용자 인증이나 봇 공격 방지를 위하여 사용하는데, 주로 웹 사이트 가입 시에 봇이 대량으로 계정을 생성하는 것을 막거나 자동 구매 봇인 '매크로'의 사용을 막기 위해 캡챠를 활용한다[9]. 따라서, 다수의 사용자가 이용하기 때문에 직관적이고 사용자가 쉽게 풀 수 있는 캡챠를 사용한다. 사용자의 행동 패턴이나 브라우저 데이터를 분석하여 봇 여부를 판단하기 때문에 사용자의 신원을 익명화하는 데에는 큰 목적이 없다. 즉, 사용자의 신원을 확인하고 사용자 행위의 합법성을 보장하기 위해 캡챠를 사용한다.

반면, 다크웹에서는 보안과 익명성 보장을 강화하기 위한 복잡한 캡챠가 사용된다. 캡챠는 봇 공격을 막는 것 외에도, 웹 사이트 자체의 익명성과 보안을 유지하는 데 중요한 역할을 한다[10]. 다크웹 사이트들을 대상으로는 크롤링 봇이나 스크래핑 툴체인들이 수사 목적 및 정보 수집을 위해 활발하게 사용되기 때문에, 불법 서비스 운영자들은 이를 회피하기 위한 목적으로 캡챠를 사용한다. 이에, 다크웹에서의 캡챠는 일반 사용자가 접근하기 어려운 방식으로 설계되며, 쉽게 우회할 수 있는 매크로의 적용이 어렵도록 복잡한 알고리즘을 사용한다. 또한, 운영자의 익명성 보장이 최우선 목적이므로, 다크웹에서는 상용 웹 서비스와의 연동을 피하려는 경향이 있다. 따라서 오픈 소스 라이브러리를 기반으로 자체 개발한 캡챠 시스템을 사용하는 경우가 다수 존재한다.

Ⅲ. 관련 연구 및 연구 목표

3.1 다크웹 크롤링 및 캡챠

자동화된 크롤링은 다크웹 연구 및 수사를 위해 필요한 핵심적인 과정인 만큼, 다양한 선행 연구에서 크롤링 기반 다크웹 데이터 수집 및 분석 방법론을 제시하고 있다. 국내 다크웹 사용자들의 언어 사용 특성을 분석한 논문에서는 'Scrapy'를 사용하여 다크웹 게시물을 크롤링했으며[11], 다크웹 불법 활동을 크롤링 및 분석한 논문에서도 크롤링을 통해 다크웹 링크를 수집했다 [12]. 이처럼, 선행 연구에서는 다크웹 사이트 링크, 불법 활동과 관련된 SNS 게시물, 다크웹 게시물 등을 수집하기 위해 일반적인 크롤링 방법을 사용하여 데이터를 수집했다.

하지만 크롤링을 통해 원활한 데이터 수집을 수행하

기 위해서는 목표 사이트가 크롤링 봇을 블락하기 위해 사용 중인 봇 탐지 방법을 우회할 수 있어야 한다. 실제로, 캐나다의 사이버 범죄와 관련된 다크웹 데이터 수집 논문에서는 속도 제한으로 인한 크롤링 어려움이 존재했으며[13], 선행 연구에서도 캡챠로 인한 크롤링 제한을 명시적으로 언급하기도 하였다[14]. 그럼에도 다크웹 생태계 내의 크롤링 봇 방지 기술 현황 및 우회 기술에 관한 선행 연구는 미비한 실정이다. 이에, 본 연구에서는 다크웹 내 히든 서비스 운영자들이 활용하고 있는 다양한 크롤링 봇 감지 기술을 실증적으로 분석하여 다크웹 데이터 수집 목적의 자동화된 크롤링 봇 구현을 위한 선결 조건을 해소하는데 기여하고자 한다.

3.2 캡챠 우회

다크웹 캡챠 파훼 방법에 관한 선행 연구에서는 단순한 텍스트 캡챠 혹은 특정 사례에만 해당되는 제한적인 조건을 가정하였거나, 최신 다크웹 캡챠 동향을 반영하지 못하는 등의 한계점이 존재한다. York Tannikos 외 1인은 다크웹 마켓 플레이스 캡챠에 대한 전반적인 개요를 작성하고 우회 방법을 구현했다[15]. 당시 연구에 따르면, 다크웹 관련 캡챠는 자체적으로 개발되며 공개적으로 사용 가능한 오픈 소스 라이브러리를 기반으로 제작된다고 설명했다. 이로 인해 Google의 ReCAPTCHA 보다 우회하기 수월했다고 말한다. 캡챠에 대해 전반적인 조사를 진행한 결과, 캡챠를 15가지 종류로 분류할 수 있었고 비교적 단순한 5개에 대한 우회 방법을 제안했다.

David Holm Audran 외 7인의 논문에서는 시계 기반의 이미지 캡챠를 분석하였다. 해당 캡챠는 EndGame V2 - Onion Service DDOS Prevention Front System의 오픈 소스로 공유되고 있으며 다크웹 상에서 범용적으로 사용되는 것을 확인했다고 언급한다[16]. 우회 모델은 ResNet 아키텍처를 기반으로 최적화된 하드웨어를 갖춘 AI 클라우드에서 훈련하여 구축했다. 노이즈로 인한 인식의 어려움을 기계 학습으로 해결하였다.

위의 두 연구에서는 특정 캡챠에 대한 우회 방법만을 논의한다. 하지만 본 연구 수행과정에서 파악한 바에 따르면, 다크웹에서는 더욱 다양하고 많은 캡챠들이 사용되며 이런 캡챠들에 대한 식별 방법론을 제시한 연구는 진행된 사례가 없다. 따라서 최신 다크웹 생태계에서 활용되는 많은 캡챠들을 수집하고 이에 대한 식별 및 우회 방법을 파악할 필요성이 존재한다.

3.3 연구 목표 및 연구 범위

다수 다크웹 사이트들은 봇, 크롤링 등을 방지하기 위해 캡챠, 세션, 임의의 URL ID 및 특정 IP 차단, 속도 제한과 같은 다중 보안 메커니즘을 사용한다[13]. 다크웹 내 데이터에 기반한 프로파일링에 대한 수요가 더 커짐에 따라 다크웹 사이트 관리자들은 더욱 보안을 강화할 것으로 예상된다. 이에 다크웹 환경을 조사하고 분석함에 있어 더 많은 어려움을 겪을 것이며 이를 해결할 방법을 찾아야 한다.

특히, 많은 다크웹 사이트들을 일일이 수동으로 확인하기는 현실적으로 어려우므로, 캡챠 여부를 확인하고 어떤 종류의 캡챠를 사용 중인지 식별하고 그에 맞는 우회 방법을 적응형으로 사용하는 자동화 기술의 개발이 필요하다. 이에 대한 첫 단계로, 본 논문에서는 다크웹 데이터 수집 관점에서 첫 장애 요소로 작용할 수 있는 크롤링 봇 차단 및 탐지 기술의 우회를 위해 최신의 다크웹 생태계에서 적용된 봇 탐지 기술 현황 분석 및 이를 우회하는 방법론을 제안한다.

Ⅳ. 다크웹 내 크롤링 봇 탐지/차단 기술

4.1 크롤링 봇 탐지/차단 기술

크롤링 봇을 탐지 및 차단하기 위해 가장 대표적으로 사용되는 캡챠는 자동화된 방식을 통해서 서버에 접근하는 것을 차단하며, 사람의 직접적인 노력을 통해서만 웹 사이트를 조작하게 만들어 크롤링 봇에 대한 보안성을 목표로 한다[20]. 표면 웹과 다크웹 모두 노이즈나 변조가 포함되어 있는 텍스트 이미지를 보고 텍스트를 입력하는 캡챠가 대표적으로 가장 많이 쓰인다. 표면 웹에서는 Google 등에서 만들어진 상용 캡챠 라이브러리를 보편적으로 사용하며, 다크웹에서는 익명성을 보장받기 위해 직접 제작한 캡챠를 많이 사용한다.

두 번째로, robots.txt를 활용하는 방법이 있다. robots.txt는 웹 페이지에 봇 배제 표준을 사용하여 접근 제한에 대한 명세를 기술한다. 각각의 user-agent에 대해 특정 경로에 접근 허용 여부를 작성하여 크롤링을 방지할 수 있으며, 해당 룰은 사이트 링크에 '/robots.txt'를 추가하여 확인할 수 있다. 상용 서비스의 경우, 대부분의 user-agent에서 알려진 크롤링 봇을 대상으로 접근을 제한한다[19]. 활용 예시로, 에어비엔비사는 크롤링 봇인 Googlebot, Bingbot 등에 대해 세부적인 경로를 작성하여 접근을 제한하고 있다. 다크웹 내에서도 Defcon과 같은 사이트에서는 AhrefsBot, CCBot, GPTBot 등을 대상으로 모든 경로의 접근을 제한하고 있다. 하지만 시그니처 기반 탐지가 갖는 한계로 인해,

알려지지 않은 방법을 통한 크롤링에 대한 대처가 불가하다는 점에서 실효성이 떨어진다는 한계가 있다. 분석 결과, 실제로 다크웹에서는 robots.txt를 사용하지 않는 경우가 많았음을 확인하였다.

마지막으로, 비정상적인 행동에 대한 접근 제한을 통해 정보 유출을 방지하는 방법이 존재한다. 예를 들어, 다크웹 내에서는 운영자가 유사하거나 동일한 IP 주소에서 상당한 수의 유사한 작업 요청을 수행하는 계정이 관찰되면 이러한 접근을 차단하거나 속도를 제한한다[17]. 대표적으로 표면 웹에서는 인스타그램 및 페이스북에서 이러한 방식을 통해 크롤링에 대처하고 있다. 다크웹의 경우, 운영자들이 수동으로 접근 로그를 확인하여 직접 접근 제한을 적용할 수 있다[18]. 이는 운영자가 직접 수시로 확인하고 차단해야 하는 번거로움이 있으므로 범용적으로 사용하기는 어렵다. 이에, 본 연구에서는 다크웹 내 캡챠 및 robots.txt 적용 현황에 초점을 맞추어 분석을 진행하였다.

4.2 다크웹 내 크롤링 봇 탐지/차단 기술 현황 분석

최신 다크웹 내 크롤링 봇 탐지 및 차단 환경을 사전 조사하고 캡챠 종류의 라벨링을 진행하기 위해 검색 엔진 'Tor Link'를 이용해 상위 15개의 링크를 샘플로 추출해 검토하였으며, 그 결과는 아래 Table 1과 같다. 공통적으로 확인한 속성은 해당 다크웹 사이트의 캡챠 존재 여부, robots.txt 여부, 사용된 캡차의 HTML 특징이다. 다수 다크웹 마켓플레이스의 경우 PGP 공개키를 제공하고 Javascript를 비활성화하도록 요구한다.

Table 1. 세 번째 열에서 확인할 수 있듯이, robots.txt를 통한 크롤링 봇 탐지/차단은 대부분 사용되지 않으

표 1. 15개의 다크웹 탐지 분석
Table 1. 15 Dark Web Detection Analysis

	Dark Web Site	Robots.txt	Captcha Type	Signature	Notes
1	APOCALYPSE	No blocking rules	Text Input	Text image and 1 input box	Enter the calculation result
2	NEXTCVV.CC	No blocking rules	(h) CAPTCHA Checkbox	Checkbox with I am human″ label	
3	CRYPTBB	No page	Text Input	Text image and 1 input box	Present on the login page
4	NEXUS MARKET	No blocking rules	Ddos form	Timer, text image and 6 input boxes	Two additional CAPTCHAs after the first one
5	DeepMarketPlace	Prohibits web crawlers from accessing URLs with page numbers, sorting option, order info, iten limits, names, and catecory filters	Time matching + Ddos form	Timer, text image and 2 select boxes	Time provided as a number
6	DREAD FORUM	No page	Ddos form	Timer, text image and 6 input boxes	
7	Tribe Seuss	No blocking rules	Time matching + Ddos form	Timer, clock image and 2 select boxes	Time provided as a clock
8	we the north	No page	URL Related	URL image and 3 input boxes	
9	NARC POLO	No blocking rules	URL Related	URL image and 2 input boxes	Text CAPTCHA also present
10	SmokersCo	No blocking rules	URL Related	URL image and 2 input boxes	Text CAPTCHA also present
11	Fish	Site closed	Image selection + Ddos form	Timer, question image and 15 selectable images	Different form of Ddos form

	Dark Web Site	Robots.txt	Captcha Type	Signature	Notes
12	cebulka	Prohibits all web crawlers from accessing admin directories, cache directories and the search page	Image selection	Selectable images	
13	Cosmos Market	No page	Text Input	Text image and 1 input box	
14	Torzon Market	Prohibits all web crawlers from accessing any pages on the site	Ddos form	Timer, text iamge and 6 input boxes	
15	ARES	No blocking rules	Image selection	Selectable images	

그림 1. 다크웹 HTML 예시
Fig. 1. Dark Web HTML Examples

며, 캡챠가 존재하면 아래 Fig.1과 같이 'captcha' 문자열이 HTML 내에서 식별되는 것을 확인할 수 있었다. 분석 결과, 다크웹 내 캡챠의 유형은 크게 텍스트, url, ddos form, image select, etc로 크게 5개의 타입으로 분류 가능했다.

4.2.1 텍스트 캡챠

텍스트 캡챠는 아래 Fig. 2과 같이 텍스트 이미지를 보고 문자 입력란에 알맞은 문자를 입력하는 구조이다. 이미지는 사이트별로 다양하며 컴퓨터가 인식하기 어렵게 노이즈나 왜곡이 포함되었다. 이미지에 포함된 문자들은 영어나 숫자가 대부분이라는 측면에서는 표면웹에서의 텍스트 캡챠와 유사한 측면이 있으나, 특수하게 중국어, 러시아어로 된 캡챠 또한 존재함을 확인하였다. 그 외에도, 단순한 문자 입력 외에도 수학 문제의 답을 입력하는 텍스트 캡챠 또한 식별되었다.

텍스트 캡챠의 공통적인 HTML 특징으로는 입력할 텍스트가 포함된 이미지와 사용자가 입력할 'text' 타입의 'input class' 요소 1개가 있다. 또한 'enter', 'in the', 'from the', 'code', 'characters' 중 하나의 문자열을 포함하고 있다는 특징이 있다. Table 2는 텍스트 캡챠의 특성을 요약한 결과이다.

텍스트 캡챠는 이미지를 보고 문자를 판단하여 입력하기만 하면 통과할 수 있다. 따라서 캡챠의 텍스트 이미지에서 노이즈 및 왜곡된 문자를 추출할 수 있다면 자동으로 캡챠를 우회할 수 있다. 이미지는 캡챠 페이지의 HTML에서 수집할 수 있고, 이미지의 텍스트는 머신 러닝을 통한 학습 모델을 활용하여 추출할 수 있다. 노이즈가 포함된 이미지를 학습을 통해 얻어내는 연구는 이미 진행된 사례가 많으며[21], 이를 활용하여 캡챠 우회를 자동화할 수 있다.

실제 우회 기능을 위한 개념 증명(Proof of Concept) 구현을 위해 이미지에서 텍스트를 뽑아낼 수 있는 Google OCR 기능을 사용하여 텍스트를 추출하였다. 대상 다크웹 사이트는 'apocalyse'로, 계산식이 포함된

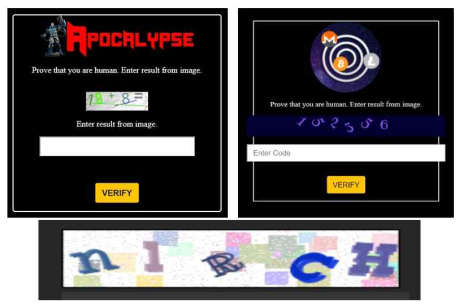

그림 2. Text 캡챠 예시들
Fig. 2. Examples of Text Captchas

표 2. 텍스트 캡챠 HTML의 패턴 특징
Table 2. Text Captcha HTML Pattern Characteristics

'captcha' Presence	O
HTML Elements	<class input ... type='text' ...> 1
Characters Included	enter, in the, from the, code, characters

이미지를 보고 캡챠 제출을 요구한다. 우회 기능 구현을 위해 이미지가 포함된 HTML 요소를 찾고, 이미지 소스가 다크웹인 점을 고려하여 안전상의 이유로 다운로드 하는 대신 이를 캡처하는 방식을 사용했다. 이미지를 얻은 후 CLAHE 대비 조절 처리를 통해 텍스트 경계 오차를 최소화한 뒤 OCR 기능을 통해 텍스트를 추출하고 사이트에 자동 입력하여 제출하도록 구현하였다. 이를 통해 전체 시도 중 16%는 우회할 수 있음을 확인하였으나, 휴리스틱한 방법론의 한계로 인해 인식하기 어려운 왜곡과 노이즈가 포함된 이미지에 대해서는 우회할 수 없었다. 따라서, 기계 학습을 통한 이미지 인식이 요구되며, 이는 CNN과 TensorFlow를 통해 숫자 네 자리 캡챠를 푸는데 최대 99.8%의 정확도를 달성한 선행 연구의 방법을 적용하여 달성 가능하다[22].

4.2.2 URL 캡챠

두 번째로, URL 기반 캡챠는 아래 Fig.3과 같이 사이트의 URL을 참고하여 빈칸을 채워 넣는 형태이다. 식별된 URL 캡챠의 경우, 현재 URL을 확인하여 빈칸으로 제공된 부분에 맞은 단어를 입력하도록 하거나 URL 전체를 붙여넣으라는 요구를 하는 형태를 보였다. 추가로, URL이 포함된 이미지 퍼즐을 맞추고 빈칸을 확인하여 입력하는 독특한 형태의 캡챠가 발견되기도 했다.

URL 캡챠에서 공통으로 나타난 HTML 구조의 특징은 아래 Table 3과 같다. URL의 빈칸 혹은 요구하는 지시문이 포함된 이미지가 존재하며 사용자의 입력을 받는 'text' 타입의 'input class'가 2개 이상 발견됐다. 해당 캡챠는 'link', 'url', 'address'와 같은 문자열 중 하나와 함께 'match', copy', 'paste' 문자열이 페어를 이루는 패턴을 보였다.

URL 캡챠는 특수한 경우가 아니라면 현재 사이트의 URL을 불러와 캡챠 우회에 사용할 수 있다. 텍스트 캡챠와 마찬가지로 빈칸 또는 지시문을 이미지에서 추출해야 하며, 이는 텍스트 캡챠의 우회 방법을 동일하게 적용하여 대응할 수 있다.

Text 캡챠 우회와 마찬가지로, URL 캡챠 또한 'We the North' 마켓 사이트를 대상으로 캡챠 우회 개념 증명을 구현하였다. 해당 사이트는 미러 사이트를 방지하기 위해 사이트 링크가 캡챠 이미지에 포함되어 있었으며, 인증을 위한 왜곡된 텍스트 또한 한 이미지에 포함되어 있었다. 이미지의 경우 캡처를 통해 확보하였으며, 흑백 전환(Grayscale Conversion) 전처리 후 Google OCR을 통해 텍스트로 변환하였다. 이후 이미지에서 변환된 캡챠 텍스트를 사이트에 자동으로 입력하고 전송

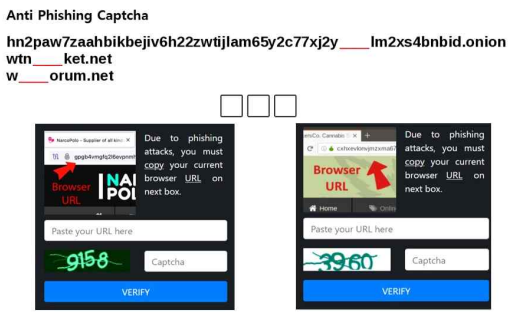

그림 3. URL 캡챠 예시들
Fig. 3. Examples of URL Captchas

표 3. URL 캡챠 HTML의 패턴 특징
Table 3. URL Captcha HTML Pattern Characteristics

'captcha' Presence	O
HTML Elements	<class input ... type='text' ...> 1
Characters Included	enter, in the, from the, code, characters
	'captcha' Presence

하도록 구현하였다. 앞의 사례와 마찬가지로 약 35%의 인식률을 보였으며, 이 또한 개념 증명 수준의 구현에 따른 텍스트 인식 오차가 원인이다. 마찬가지로 기계 학습을 통한 개선을 고려했을 때, URL 캡챠 유형을 식별할 수 있다면 우회 성공률 또한 향상시킬 수 있다.

4.2.3 DDoS form 캡챠

DDoS form 캡챠는 일반 표면 웹에서와 달리 다크웹 내 다수의 사이트에서만 식별된 형태로, 제한된 시간 내에 이미지에 보이는 문자를 각 입력란에 입력해야 한다는 특성을 가진다. 아래 Fig. 4와 같이 이미지에는 노이즈와 왜곡이 포함되었고 각 입력란을 선택하면 입력해야 할 문자 이미지로 이동하여 텍스트를 보여준다. 해당 캡챠 페이지는 모두 리다이렉션(redirection)되어 로딩되었으며, 캡챠 페이지와 함께 1분 내외의 제한 시간이 설정되어 있었다.

Table 4와 같이, 해당 캡챠에 공통적인 HTML 특징으로는 'text' 타입의 'input class' 요소가 6개 존재했다. 또한 'ddos form', 'ddos', 'expire', 'timer' 문자가 확인되었다. 이종의 사이트로부터 식별한 DDoS form 캡챠들은 사용된 변수 이름만 조금씩 다를 뿐, 유사한 HTML 구조를 보인다는 점 또한 확인하였다.

DDoS form 캡챠는 자동 우회를 어렵게 하도록 다양한 방법을 사용한다. 첫 번째로, 여러 개의 이미지를

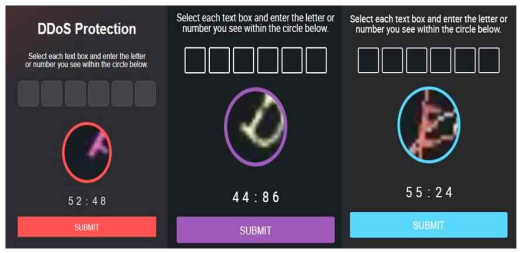

그림 4. DDos Form 캡챠 예시들
Fig. 4. Examples of DDoS Form Captchas

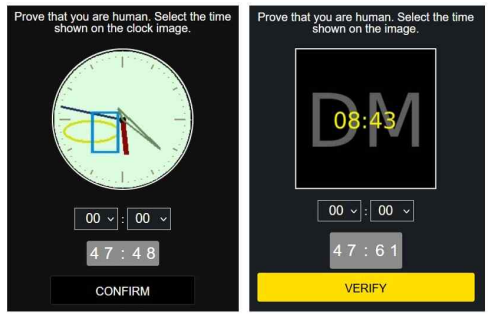

그림 5. DDos Form Clock 캡챠 예시들
Fig. 5. Examples of DDoS Form Clock Captchas

표 4. DDos Form 캡챠 HTML의 패턴 특징
Table 4. DDoS Form HTML Pattern Characteristics

'captcha' Presence	O
HTML Elements	<class input ... type='text' ...> 6
Characters Included	ddos_form, ddos, expire, timer

활용하여 사이트에 접속할 때마다 다양한 캡챠 이미지로 리다이렉션 시킨다. 따라서, 우회를 위해서는 활용되는 이미지의 경우의 수를 모두 수집해야 한다. 두 번째로, 한 이미지에 존재하는 여러 텍스트에 대한 입력을 요구한다. 노이즈 및 왜곡을 제거하고 텍스트를 추출한다 해도, 6개의 입력란이 어떤 텍스트를 보여줄지 알 수 없으므로 모든 텍스트를 추출하여 6개로 만들 수 있는 모든 조합을 시도해 보아야 한다. 세 번째로, 위 모든 행위를 1분이라는 제한 시간 내에 성공해야 한다.

해당 캡챠에 등장하는 이미지를 충분히 학습시킨 모델과 6가지 이미지를 빠르게 캡처할 수 있는 스크립트가 존재한다면 1분 안에 캡챠를 해결하는 것을 기대해 볼 수 있다. 해당 사이트의 캡챠는 상당한 왜곡이 포함되어 있으므로, 앞선 사례들과 같이 단순한 이미지 전처리 및 OCR 기능을 통해서는 우회할 수 없었다. 다만, 위 조건을 충족시키면서 직접 원본의 캡챠 이미지를 수집하고, 문자를 개별적으로 분리하여 학습한 뒤 획득한 모델을 통해 추론을 수행한다면 우회할 수 있다[23].

또한, 텍스트 및 이미지 캡챠 형식을 응용하여 변형한 캡챠도 식별되었다. 텍스트 입력 대신 시계 이미지를 제공하여 시간을 맞추도록 하는 형태로, 아래 Fig. 5와 같이 캡챠의 시계 이미지에는 시계와 상관없는 도형이나 다른 색의 시침/분침을 포함시켜 봇이 이미지를 인식할 수 없도록 한다. 텍스트 입력이 아닌 드롭다운을 통

표 5. 변형 DDoS Form 캡챠 HTML의 패턴 특징
Table 5. Modified DDoS Form Captcha HTML Pattern Characteristics

'captcha' Presence	O
HTML Elements	<select class ...> 2
Characters Included	clock, image, select, time
Additional Information	The class above has <option value='00'~ '11'>

한 선택을 통해 시간을 입력하며, 아날로그나 디지털 형태로 시간을 보여준다.

위 시계 형태의 변형 DDoS form 캡챠 또한 일반적인 DDoS form과 HTML 구조와 유사하다. Table 5에 나타나듯이, 고유한 특징으로는 '00'부터 '11'이상의 값을 가지는 select class를 공통으로 포함, 시와 분을 맞추기 위해 2개 존재한다. 또한 'clock', 'image', 'select', 'time'의 단어가 모두 존재하는 것이 확인되었다.

DDos form(clock) 아날로그 이미지의 경우 선행연구에서 설계한 ResNet50 기반 모델 적용을 통해 우회할 수 있다[16]. 해당 캡챠는 720가지의 가능한 시간 중 정확한 시각을 판별하는 문제를 해결하기 위해 캡챠 생성 코드를 활용하여 자동으로 합성 데이터를 생성 및 라벨링하였으며, Python과 Keras API를 사용하여 미학습 ResNet50 모델을 커스터마이징하여 학습을 진행하였다. 해당 모델의 경우는 정확도 96.83%로 캡챠를 높은 성공률을 보이기에, 본 제안 기술과 함께 활용될 경우 해당 캡챠 유형 또한 식별 및 우회 가능하다.

4.2.4 Image select 캡챠

Image select 캡챠는 주어진 이미지에서 요구되는 특정 이미지(아이콘)를 클릭하는 방식이다. 확인된 캡챠

들이 사용하는 이미지들은 모두 다르지만, 아래 Fig.6과 같이 여러 개의 원 중에 끊긴 원을 선택하는 방법은 동일하게 나타났다. 이미지에 나타난 끊긴 원을 클릭하면, 해당 좌표가 서버로 전송된다. 이를 통해 사용자가 봇인지를 판단한다.

해당 캡챠는 아래 Table 6과 같이 공통적으로 'action' 속성을 가진 'form'내에 'image' 타입의 'input'이 있으며 'circle', 'onion', 'sliced', 'cliick', 'cut', 'broken' 문자열을 포함하며, 사용자가 이미지의 특정 부분을 클릭하면 해당 좌표가 form 데이터에 포함되어 서버로 전송된다.

Image select 캡챠는 끊어진 원이 그려진 부분을 찾아 클릭해야 하므로, 이미지를 보고 판단할 수 있어야 한다. 하지만 이미지에 노이즈가 많이 추가되어 있으므로 분석이 어려울 수 있다. 해당 캡챠를 자동으로 우회하기 위해서는 이러한 이미지들을 학습을 통해 끊긴 원의 지점을 정확히 찾아낼 수 있어야 한다.

해당 캡챠 타입은 Ddos form (default) 타입과 달리 기계 학습을 사용하더라도, 끊긴 원과 겹쳐진 원의 차이를 고려하여 훈련하도록 하는 것은 상당한 어려움이 따를 것이다. 이 경우, '2Captcha' 서비스 사용을 고려해 볼 수 있다. 2Captcha는 기계 학습 모델로 해결하기 어려운 캡챠를 비용을 지불하고 사람이 대신 풀어주는 서비스이다. 이 플랫폼에는 실제 작업자들이 등록되어 있으며, 사용자가 업로드한 캡챠를 보고 직접 풀이하며, 작업자가 입력한 정답은 API를 통해 사용자에게 반환된다. 특히 Image select과 같은 클릭 기반 캡챠의 경우, 2Captcha 서버는 작업자의 클릭 위치를 좌표 값으로 변환하여 제공한다. 이를 통해 사용자는 직접 클릭할 필요 없이 Selenium과 같은 자동화 도구를 활용하여 정확한 좌표를 클릭하는 방식으로 캡챠를 해결할 수 있다.

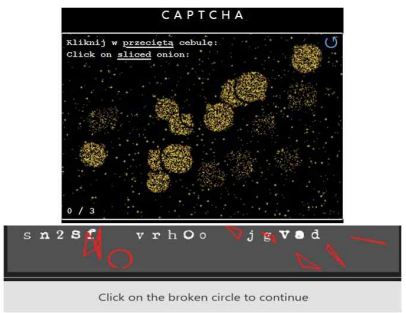

그림 6. Image Select 캡챠 예시들
Fig. 6. Examples of Image Select Captchas

표 6. Image Select 캡챠 HTML의 패턴 특징
Table 6. Image Select Captcha HTML Pattern Characteristics

'captcha' Presence	O
HTML Elements	<form action ... method='POST' ....>
Characters Included	circle, onion, sliced, click, cut, broken
Additional Information	Above <form> has <input type='image' ...>

4.2.5 기타 유형(Miscellaneous)

앞에서 분류된 캡챠 이외에도 추가로 식별된 유형들이 존재하였는데, 먼저 상대적으로 손쉽게 우회 가능한 상용 캡챠인 'hcaptcha'가 적용된 경우도 존재했다. 해당 다크웹 사이트의 경우 불법 행위와 관련되지 않아 익명성이 보장되지 않아도 되는 사이트임을 식별하였다. 그 외에도 Fig. 8과 같이 이미지 퍼즐을 맞추는 형태, 주어진 문자와 동일한 문자를 선택하는 형태, 제공된 이미지에서 다른 아이콘을 클릭하는 형태 등 여러 캡챠를 동시에 사용하는 경우로 일부 존재했다. 해당 캡챠들은 가지 수가 적어, 별도 유형으로 구분하지 않고 etc로

그림 7. 다양한 캡챠
Fig. 7. Miscellaneous Captcha

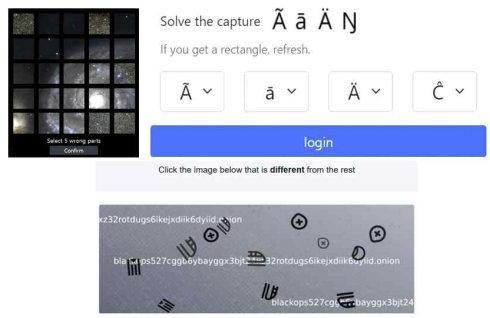

그림 8. 다양한 캡챠
Fig. 8. Miscellaneous Captcha (Hybrid)

라벨링하여 분류하였다.

4.3 다크웹 캡챠 유형별 시그니처 추출

자동으로 캡챠를 분류하기 위한 프레임워크 개발을 위해, 앞서 분석한 캡챠 타입들의 특징을 기반으로 분류 로직 코드를 구현하였다. 기본적으로 로직 코드는 먼저 캡챠 사이트의 HTML 태그들을 확인하여 검사한다. 분류 전 HTML 구조 내 구문 일치 과정을 통해 캡챠의 존재 유무를 확인하고, 캡챠 타입 분류는 캡챠의 특징이 명확한 것부터 우선순위를 두어 휴리스틱하게 검사하도록 구성하였다. 태그 분석을 통해, 우선 순위는 ddos form(clock)부터 ddos form(default), image select, URL, text 캡챠 순으로 정하였으며, 나머지 미분류 캡챠들은 etc로 분류된다.

각 캡챠 별로 식별 알고리즘을 개발하였으며, 그 예시 중 하나인 시계형 DDoS form 캡챠 탐지 알고리즘은 Fig. 9와 같다. 먼저 Ddos form(clock) 캡챠의 경우, ‘option value’가 ’00‘~’11‘의 일련의 숫자가 포함되는 ’select class’를 확인하며, 해당 ‘class’요소는 두 개 이

Algorithm check_ddos_form_clock

```
required_values ← {strings "00" to "11"}
matching_select_count ← count of select elements that contain all of required_values
if matching_select_count < 2 then
    return "no"
else
    return "yes" if "image", "ddos_form" in HTML else "no"
end if
```

그림 9. DDos_form(clock) 캡챠 탐지 알고리즘
Fig. 9. DDos_form(clock) Captcha Detection Algorithm

Algorithm check_url_captcha

```
condition_1, condition_2 ← False, False
for each p in all p_tags in HTML do
    if "link", "url", "address" in p_tags then
        condition_1 ← True
    end if
    if "copy", "paste", "match" in p_tags then
        condition_2 ← True
    end if
    if condition_1 and condition_2 and len(input boxes with type='text') ≥ 1 then
        return "yes"
    end if
end for
return "no"
```

그림 10. URL 캡챠 탐지 알고리즘
Fig. 10. URL Captcha Detection Algorithm

상 존재해야 한다. 그리고 [’image’, ’select’, ’time’, ddos_form’] 리스트 내부의 모든 문자를 포함해야 한다. 마지막으로 ddos form(default) 캡챠의 문자 리스트 중 한 단어라도 포함해야 한다.

또 다른 예시로, URL 캡챠는 캡챠와 관련된 단어를 확인하기 위해 아래 Fig.10과 같이 <p> 태그 내부의 문자를 검토한다. 추가로, [’link’, ’url’, ’address’] 리스트 중 하나라도 존재해야 하며, 동시에 [‘match’, ’copy’, ’paste’] 리스트에서도 한 단어를 포함해야 한다. 마지막으로, ’text’ 타입의 ‘input class’를 한 개 이상 가지고 있어야 한다.

V. 다크웹 캡챠 수집 및 분류 프레임워크

다크웹 환경에서 캡챠를 얼마나 많이 사용하고 어떤 캡챠를 주로 사용하는지에 대한 통계적 분석을 위해, 본 논문에서는 캡챠 분류를 위한 데이터 수집과 더불어 자동화된 다크웹 캡챠 수집 및 분류 프레임워크를 개발하였다. 주요 기능들은 크게 3가지로 요약할 수 있으며, 먼저 아카이빙 사이트에서 onion 링크들을 수집하는 onion_cralwer 모듈이 존재한다. 또한, 프록시를 이용하여 수집한 사이트들의 랜딩 HTML을 가져와 데이터베이스화 하는 html_crawler 모듈, 수집된 HTML 데이터베이스를 바탕으로 식별 알고리즘을 통해 캡챠를 분류하는 captcha_checker 모듈이 존재한다. Fig. 11은 프레임워크의 동작 흐름 및 구성 모듈을 나타낸 것이다.

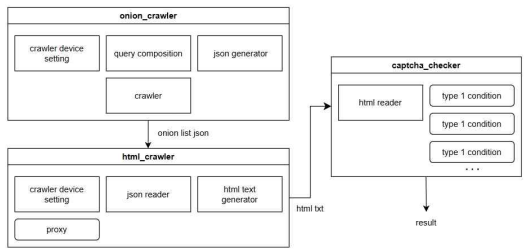

그림 11. 프레임워크 설계도
Fig. 11. Framework design

5.1 URL 수집 (onion crawler 모듈)

링크 수집 모듈은 아카이빙 사이트에서 시드(seed)가 되는 onion 링크를 자동으로 수집한다. 사용자에게 아카이빙 사이트에 질의할 키워드를 입력받아서 각 키워드와 페이지 수를 쿼리로 요청한다. 이렇게 수집된 URL들은 JSON 파일 형태로 저장되어 HTML 데이터베이스화에 사용된다. JSON 파일의 구조는 인덱스인 ‘id’, 실제 캡챠 유형 라벨인 ’captcha’, 사이트명

'name', 사이트 링크인 'url', 아카이빙 사이트에 검색한 키워드 'keyword', 수집된 페이지 정보인 'page'로 구성된다.

5.2 HTML 데이터베이스화 (html cralwer 모듈)

HTML 데이터베이스화 기능은 수집한 onion URL에 접속하여 로딩된 첫 랜딩(landing) 페이지의 HTML을 크롤링하여 데이터베이스 형태로 저장하는 기능이다. 이때, 모든 URL에 대한 HTML 원시 데이터 또한 함께 저장하며, 이는 캡챠 분류 알고리즘의 입력으로 함께 사용된다. onion 사이트에 접속하기 위해서는 proxy를 사용해야 하는데, 이를 위해 Selenium에 'socks5'를 옵션으로 주어 설정했으며, 빠른 처리를 위해 멀티 스레드를 통해 병렬 처리하도록 구현하였다.

5.3 캡챠 분류 (captcha checker 모듈)

캡챠 분류 기능은 캡챠 존재 여부를 판단 후, 존재한다면 캡챠 타입을 분류한다. 캡챠를 분류하기 위해서 앞선 단계에서 정의한 캡챠 타입 별 시그니처를 활용하여 판단하는 알고리즘을 기반으로 동작한다. 이때, 휴리스틱 알고리즘의 한계로 인한 오탐을 최소화 하기 위해, 고유한 특성을 보유한 캡챠 유형들을 우선적으로 검사하여 어떤 타입의 캡챠인지 식별하도록 구현하였다.

Ⅵ. 성능 평가

6.1 수집 결과

'Tor Link' 아카이빙 사이트에서 키워드를 기반으로 onion URL을 수집한 결과 125개의 사이트를 확보하였으며, 수집 중, 캡챠가 사이트의 로그인 페이지나 등록 페이지에 존재하는 경우는 URL을 매뉴얼하게 수정하여 캡챠가 있는 부분으로 업데이트해주었다. Fig. 12은 수집한 JSON 포맷의 예시를 나타낸다. 수집한 사이트의 URL을 기반으로 HTML을 분석한 결과, 125개 중 5개의 URL이 오프라인으로 수집 불가한 상태여서 120개의 URL에 대해 캡챠 분류 모델을 통해 분류를 진행했다.


```
[
    {
        "id": 1,
        "captcha": "None",
        "name": "Dimension X",
        "url": "http://tx3rlqycg7wais576gsuirtpuuvkxewpur2m3bkn4t5kludxw7sq4gad.onion/",
        "keyword": "weed",
        "page": 1
    },
    {
        "id": 2,
        "captcha": "url",
        "name": "SMOKERSCO - CANNABIS SHOP IN DARK WEB",
        "url": "http://slorebzyk76dhafdwvzdnqmql2h5vwxc23dt5lx4mlzxs2kv3uteliad.onion/",
        "keyword": "weed",
        "page": 1
    },
    {
        "id": 3,
        "captcha": "None",
        "name": "MediMart DrugStore",
        "url": "http://ka6n6npqxdk243ncrefuezozchb5rf3hqwm2fgtdmhwvcv7ogmqfqjid.onion",
        "keyword": "weed",
        "page": 1
    },
```


그림 12. JSON 형태의 수집된 캡챠 예시
Fig. 12. Captcha Collection JSON Example

6.2 평가 지표 및 프레임워크 정확도

캡챠 로직 검사 결과의 정확도를 확인하기 위해 수동으로 확인한 실제 캡챠 결과와 프레임워크가 분류한 결과를 비교했다. 캡챠 분류의 결과 라벨은 'text', 'url', 'ddos_form(default)', 'ddos_form(clock)', image_select', 'etc', none'을 가진다. 이는 그래프를 표현함에 있어, 7가지의 class로 다중 분류에 해당한다. 다중 클래스는 OvR의 문제이기 때문에 자신의 class에는 Positive, 나머지 class는 모두 Negative로 하여 계산을 진행한다. 자신의 class 정답과 예측이 동일할 경우 True, 정답과 예측이 불일치 할 경우를 False로 판단한다. 이때의 각 캡챠 유형별 재현율(Recall), 정밀도(Precision), 정확도(Accuracy)를 계산했다.

$$Accuracy = \frac{TruePusitives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \quad (1)$$

$$Recall = \frac{TruePasitives}{TruePositives + FalseNegatives} \quad (2)$$

$$Precision = \frac{True\ \ Positives}{True\ \ Positives + False\ \ Positives} \quad (3)$$

먼저 120개의 URL을 기반으로 캡챠의 분류를 진행한 결과, 44개의 URL에서 캡챠가 존재함을 알 수 있었

표 7. 재현율과 정밀도
Table 7. Recall and Precision

	text	URL	ddos form (default)	ddos form (clock)	image	etc	none	AVG
Recall	83.33%	100%	71.42%	100%	100%	87.5%	97.33%	91.36%
Precision	83.33%	80%	100%	100%	100%	87.5%	96.05%	92.41%

다. 다시 말해, 다크웹 환경의 캡챠 사용률은 36.67%로 데이터 수집에 유의미한 영향을 미칠 수 있는 수치이다.

캡챠 유형별 재현율과 정밀도를 기반으로 계산한 평균 재현율은 Table 7과 같고, JSON 파일의 'captcha'와 비교하여 분류 로직이 판단한 캡챠 타입이 라벨 별로 일치하는지 식별하기 위한 오차행렬(confusion matrix)는 아래 Fig. 13과 같다. 87.12%, 평균 정밀도는 91.8%, 정확도는 92.5%이다. 이 통계치를 통해 알 수 있듯이, 본 연구의 프레임워크는 높은 일관성을 보인다. 최종적으로 캡챠 분류의 정확도는 93.3%로 높은 성능 또한 보이는 것을 확인할 수 있었다.

다크웹 내 캡챠 유형에 대한 자동화된 다중 분류를 수행한 연구가 부재하여, 선행 연구[16,22,24] 내 특정 캡챠 유형에 대한 우회 성능과 간접적으로 성능 비교를 수행하였다. 약 96%의 텍스트 캡챠 우회 성능 및 84%의 이미지 캡챠 우회 성능을 가진 선행연구와 비교했을 때, 이에 준하는 높은 성능을 보이기에 실효성을 가진다고 볼 수 있다. 특히,

선행 연구의 우회 기법과 제안한 캡챠 유형 분류 기술을 함께 활용할 수 있다. 또한, 선행 연구[24]에서 사람의 수동 평균 캡챠 풀이 성능이 약 87%이기에, 제안한 프레임워크를 활용하면 이보다 효율적으로 캡챠 유형을 분류할 수 있음을 시사한다.

캡챠가 존재하는 URL에서 유형에 따른 캡챠의 분포는 아래 Fig. 14과 같다. text 캡챠는 18개, image select 캡챠의 경우는 6개, URL과 ddos form(default) 캡챠는

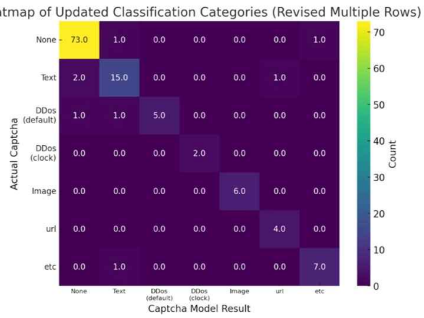

그림 13. 오차행렬
Fig. 13. Confusion matrix

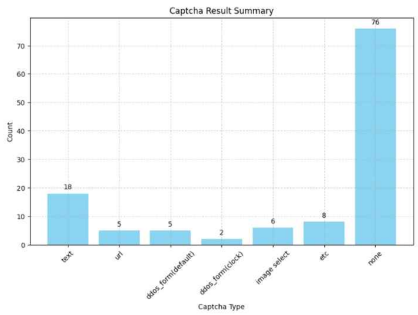

그림 14. 캡챠 분류 결과
Fig. 14. Captcha classification results

5개, ddos form(clock)은 2개, 마지막으로 etc의 경우는 8개로 분류됨을 확인하였다. 2024년 기준 다크웹에서 가장 빈번히 사용되는 캡챠는 text 기반 캡챠로, 캡챠가 적용된 페이지 중 약 41%가 이를 사용함을 확인하였다.

6.3 오차 분석

전체 120개의 URL에서 캡챠 분류기가 정확하게 판단하지 못한 8개의 결과를 토대로 상대오차를 계산하면 6.67%의 상대 오차율을 보이는 것을 알 수 있으며, 위 Table 8은 미스매치가 발생한 라벨들의 결과를 나타낸다. 오차가 발생한 원인은 주로 HTML에 있다. 사이트의 리다이렉팅 문제나 로딩 문제로 인해 HTML을 제대로 불러오지 못한 경우에는 정확한 캡챠 타입의 분류가 어렵다. ddos form의 경우에는 모두 이 경우 해당하며, 리다이렉팅 페이지의 HTML로 인해 text로 분류될 수 있다. 또한, 캡챠 이외의 다른 목적으로 사용되는 'input class'가 존재할 경우 잘못된 결과를 출력한다. 예를 들어 search, ID/Password 등과 관련된 박스로 사용되는 경우가 계산되는 경우가 존재한다. 그리고 각 캡챠 타입에 해당되는 키워드들이 다른 캡챠 타입에도 존재할 경우, 상위 로직에 필터링되며 예외의 특정 키워드가 사용되어 분류가 불가능할 수 있다.

표 8. 오차 케이스들
Table 8. Error Case Studies

actual captcha	None	text	ddos form (default)	etc	ddos form (default)	None	text	text
captcha sorting machine	text	URL	text	text	None	etc	None	None

Ⅶ. 논의 및 고찰

본 연구의 한계점 중 하나로, 수집된 onion 사이트의 모수가 부족하다는 점이 존재한다. 이는 다크웹 운영자들은 수사의 추적을 피하기 위해 주기적으로 링크 주소를 변경하기 때문으로, 링크 리스트를 계속 새롭게 업데이트할 필요가 있다. 추가로, HTML 수집에 상당히 많은 시간이 소요된다. 리다이렉팅을 수행하는 사이트들, 혹은 봇을 막기 위한 수단으로써 대기 시간을 걸어 놓는 사이트들로 인해 랜딩 페이지를 보기 위해서는 평균 40초의 대기가 필수적으로 필요했다. 해당 시간이 필요한 사이트와 아닌 사이트에 대해 적재적소로 대기 시간을 적용하지 못했다.

본 연구에서는 랜딩 페이지에 한해 캡챠를 수집하였는데. 만약 여러 개의 캡챠가 다중으로 적용된 사이트가 존재한다거나 다른 기능의 페이지에 캡챠가 존재하는 경우 자동 분석이 어렵다는 한계점 또한 존재한다. 실제로 etc로 분류된 케이스 내에는 아래 Fig. 15와 같이 총 3개의 캡챠가 다중 적용된 사례가 존재함을 확인했다. 추후 이러한 다중 사례에 대해서도 라벨을 추가한 뒤, 캡챠에서 제공하는 이미지를 인공지능 학습을 통해 분석하여 컴퓨터가 이해할 수 있는 문자 형태로 추출해 낼 수 있다면, 캡챠를 우회하는 자동화된 캡챠 우회 시스템을 구축할 수 있을 것으로 기대된다. 추후 인공 지능 기술을 활용하여 이미지에서의 텍스트 추출, 각 유형별 캡챠를 자동으로 우회하여 데이터를 수집하는 프레임워크로 확장할 계획이다.

그림 15. 다중 캡챠 예시
Fig. 15. Multi-layered Captcha Example

Ⅷ. 결 론

본 논문에서는 불법적인 암시장의 근간이 되는 다크웹 내 수사를 위한 데이터 수집을 어렵게 하는 주요 원인인 다크웹 내 캡챠의 최신 적용 현황을 분석하고 각 유형별로 자동화된 수집 및 분류를 수행하는 프레임워크를 제안하였다. 분석 결과, 전체 onion 사이트 중 약 40%가 크롤링 봇 차단을 위해 캡챠를 사용하고 있었으며, 사용되는 캡챠 중 가장 많은 비중을 차지했던 것은 text 캡챠임을 확인하였다. 본 연구가 제안한 프레임워크의 정확도는 93.33%로 분석하고자 하는 캡챠에 대해서는 준수한 성능을 보여주었다. 사례 연구를 통해, 노이즈나 왜곡이 들어간 텍스트 이미지, 요구하는 특정 아이콘이 존재하는 이미지, 노이즈가 추가된 시계 이미지, 빈칸이 추가된 URL 이미지 등 표면 웹과 차별화된 특성들이 존재했으며, 다크웹 상에서는 이러한 고유 특성을 기반으로 한 시그니처 추출을 통해 캡챠 우회 기술의 적용이 필요함 또한 확인하였다. 다크웹 링크 수집 시, 캡챠 적용 여부에 대한 파악을 위해 제안한 프레임워크가 활용될 것으로 기대하며, 캡챠에서 제공하는 이미지를 추출, 분석 후 인공 지능 기반 캡챠 우회 로직과 연동될 경우 다크웹 수사의 효율성이 향상될 것이다.

References

[1] B. Won, "*Dark web drug transactions increase 13-fold over 5 years*(2023)," Retrieved Dec. 5, 2024, from https://m.boannews.com/html/detail.html?idx=122230.

[2] E. Jardine, "The dark web dilemma: Tor, anonymity and online policing," *Global Commission on Internet Governance Paper Series*, no. 21, pp. 1-24, Sep. 2015. (https://doi.org/10.2139/ssrn.2667711)

[3] D. Pascale, G. Cascavilla, D. A. Tamburri, and W. J. Van Den Heuvel, "CRATOR: A CRAwler for TOR: Turning dark web pages into open source INTelligence," in *Computer Security - ESORICS 2024*, pp. 144-161, The Hague, Netherlands, Sep. 2024. (https://doi.org/10.1007/978-3-031-70890-9_8)

[4] Y. Wang, B. Arief, and J. Hernandez-Castro, "Analysis of security mechanisms of dark web markets," *EICC 2024*, pp. 1-8, Jun. 2024. (https://doi.org/10.1145/3655693.3655700)

[5] J. Pastor-Galindo, F. G. Mármol, and G. M. Pérez, "On the gathering of Tor onion addresses," *Future Generation Computer Syst.*, vol. 145, pp. 12-26, Aug. 2023. (https://doi.org/10.1016/j.future.2023.02.024)

[6] Tor Link, "*About Tor link*," Retrieved Dec. 5, 2024, from https://tor.link/about

[7] Agnė Augustėnė, *What is Captcha*(2022), Retrieved Nov. 20, 2024, from https://nordvpn.com/ko/blog/captcha-meaning/

[8] G. Cho, J. Choi, and Y. Kim, "CAPTCHA trends in terms of security and usability," *Rev. KIISC*, vol. 27, no. 1, pp. 47-54, 2017. (https://doi.org/10.5469/jkma.2017.40.2.112)

[9] H. Sohn, "What is the "*"Macro" program used to manipulate comments* (2018)," Retrieved Nov. 20, 2024, from https://www.joongang.c o.kr/article/22541596.

[10] K. Csuka, D. Gaastra, and Y. de Bruijn, "*Breaking CAPTCHAs on the dark web*," University of Amsterdam, System & Network Engineering Report, 2018. (https://rp.os3.nl/2017-2018/p62/report.pdf)

[11] Y. Lee, D. Yim, and Y. Lee, "Analyzing the language usage characteristics of Korean dark web users," in *Annual Conf. Human and Language Technol.*, pp. 397-402, Kyongju, Korea, 2022. (https://doi.org/10.5469/CFKO2022264553468 87)

[12] M, A. Hadi, R. M. Alaidi, H. T. S. Al_airaji, I. A. Alrikabi, and S. H. A. Aljazaery, "Dark web illegal activities crawling and classifying using data mining techniques," *Int. J. Interactive Mobile Technol.* (*iJIM*), vol. 16, no. 10, pp. 122-139, May 2022. (https://doi.org/10.3991/ijim.v16i10.30209)

[13] E. Crowder, and J. Lansiquot, "Darknet data mining: A canadian cyber-crime perspective," *arXiv preprint arXiv:2105.13957*, May 2021. (https://doi.org/10.48550/arXiv.2105.13957)

[14] A. Baravalle, M. M. S. Lopez, and S. W. Lee, "Mining the dark web: Drugs and fake IDs," in *Proc. 2016 IEEE 16th ICDMW*, pp. 350-356, Barcelona, Spain, 2016. (https://doi.org/10.1109/ICDMW.2016.0056)

[15] Y. Yannikos and J. Heeger, "Captchas on darknet marketplaces: Overview and automated solvers," in *Proc. Electr. Imaging Symp.*, pp. 1-6, San Francisco, USA, Jan. 2024. (https://doi.org/10.2352/EI.2024.36.4.MWSF-330)

[16] D. Audran, M. Andersen, M. Hansen, M. Andersen, T. Frederiksen, K. Hansen, D. Georgoulias, and E. Vasilomanolakis, "Tick tock break the clock: Breaking CAPTCHAs on the darkweb," in *Proc. 19th Int. Conf. SECRYPT 2022*, pp. 357-365, Lisbon, Portugal, 2022. (https://doi.org/10.5220/0011273300003283)

[17] A. Ouaddah, H. Mousannif, A. A. Elkalam, and A. A Ouahman, "Access control in the internet of things: Big challenges and new opportunities," *Computer Netw.*, vol. 112, pp. 237-262, Jun. 2016. (https://doi.org/10.1016/j.comnet.2016.11.007)

[18] N. Zhang, M. Ebrahimi, W. Li, and H. Chen, "Counteracting dark web text-based CAPTCHA with generative adversarial learning for proactive cyber threat intelligence," *ACM TMIS*, vol. 13, no. 2, pp. 1-21, Mar. 2022. (https://doi.org/10.1145/3505226)

[19] Y. Sun, Z. Zhuang, and C. L. Giles, "A large-scale study of robots.txt," in *Proc. 16th Int. Conf. World Wide Web*, pp. 1123-1124, Banff, Alberta, Canada, 2007. (https://doi.org/10.1145/1242572.1242726)

[20] R. Gossweiler, M. Kamvar, and S. Baluja, "What's up CAPTCHA? A CAPTCHA based on image orientation," in *Proc. 18th Int. Conf. World Wide Web*, pp. 841-850, New York, NY, 2009. (https://doi.org/10.1145/1526709.1526822)

[21] Y. Cai, X. Hu, H. Wang, Y. Zhang, H. Pfister, and D. Wei, "Learning to generate realistic noisy images via pixel-level noise-aware adversarial training," *NIPS*, vol. 34, pp. 3259-3270, 2021. (https://doi.org/10.48550/arXiv.2204.02844)

[22] Github, "*Captcha-tensorflow*," Retrieved Feb. 7, 2025, form https://github.com/JackonYang/captcha-tensorflow

[23] N. Tariq, F. A. Khan, S. A. Moqurrab, and G. Srivastava, "CAPTCHA types and breaking

techniques: Design issues, challenges, and future research directions, ACM computing surveys," *ACM Comput. Surv.*, 2023. (https://doi.org/10.48550/arXiv.2307.10239)

[24] H. Nejati, N. M. Cheung, R. Sosa, and D. C. I. Koh, "DeepCAPTCHA: An image CAPTCHA based on depth perception," in *Proc. 5th ACM Multimedia Syst. Conf.* pp. 81-90, Mar. 2014. (https://doi.org/10.1145/2557642.2557653)

유 은 선 (Eunseon Yu)

2025년 2월 : 성신여자대학교 융합보안공학과 졸업

<관심분야> 다크웹, 컴퓨터 보안

[ORCID:0009-0003-9006-8083]

박 규 나 (Gyuna Park)

2021년 2월~현재 : 성신여자대학교 융합보안공학과

<관심분야> 다크웹, 컴퓨터 보안

[ORCID:0009-0009-4674-4578]

백 서 이 (Seo-Yi Baik)

2022년 2월~현재 : 성신여자대학교 융합보안공학과

<관심분야> 다크웹, 컴퓨터 보안

[ORCID:0009-0005-3171-3681]

김 성 민 (Seongmin Kim)

2012년 2월 : 한국과학기술원 전기 및 전자공학과 공학사

2014년 2월 : 한국과학기술원 전기 및 전자공학과 석사

2019년 2월 : 한국과학기술원 정보보호대학원 박사

2020년 9월~현재 : 성신여자대학교 융합보안공학과 조교수

<관심분야> 클라우드 컴퓨팅, 시스템 보안

[ORCID:0000-0002-8183-0641]