SCA: 안드로이드 악성 앱 탐지의 개념 드리프트 대응을 위한 보안 기반 적응 모듈

2026 한국컴퓨터종합학술대회

권나희, 노경민, 조원빈, 박수현, 조성제
Computer Security & OS Lab.

2026.06.24

Contents

PART 01

Introduction

배경과 문제점

Background

Android 악성 앱 위협의 실제 규모

악성 앱 증가 추세

- 2023년 신규 Android 악성 앱 380만 건 이상 보고
- 전세계 모바일 악성코드 97%가 Android 대상
- 월평균 33만 건 이상의 신규 악성 앱 탐지
- 공식 앱마켓 외 sideloading을 경로로 배포 확산

탐지 시스템의 실제 운영 현실

- Google Play Protect: 매일 1,250억 앱 스캔
- 탐지 모델은 특정 시점 데이터로 학습됨
- 새로운 악성 앱 변종은 기존 패턴과 상이
- 시간이 지날수록 탐지율이 저하되는 현상 관찰

→ 분포 변화 속에서도 성능을 유지하는, 안정적이고 지속 가능한 탐지 체계 필요

출처: Kaspersky Security Bulletin 2023, Google Play Protect Report 2024, Symantec Internet Security Threat Report

Problem

문제 정의: Concept Drift란 무엇인가

Concept Drift

시간이 지남에 따라 입력 데이터의 통계적 분포가 변화하여, 학습 당시 적합했던 모델의 결정 경계가 더 이상 유효하지 않게 되는 현상

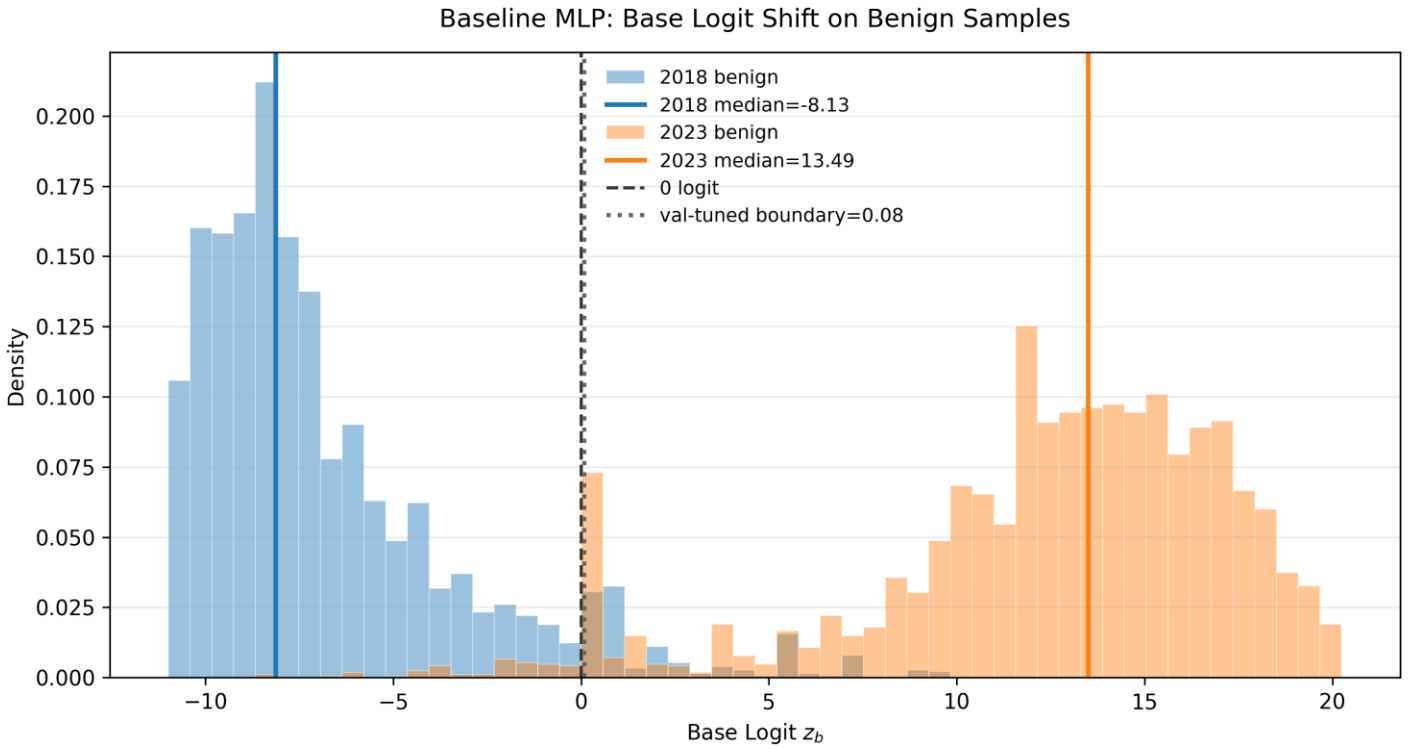
데이터 분포가 학습 분포에서 벗어날 때 모델의 정확도 뿐만 아니라 예측 신뢰도가 함께 저하되며, 이는
모델의 Concept Drift에 대한 강건성을 약하게 만든다. [Ovadia et al., NeruIPS 2019]

Problem

문제 정의: Logit shift란 무엇인가

Logit shift

Concept Drift로 인해 동일 클래스의 Base Logit 분포가 학습 시점과 다르게 이동하는 현상



Contribution

At a glance: 우리 방법은 이 문제를 어떻게 해결할 것인가

1. 특정 DL 모델에 종속되지 않는 Logit 보정 모듈

기존 탐지 모델을 변경하지 않고 Logit 보정 모듈을 추가하여 logit 보정 수행

2. Logit shift 보정을 통한 드리프트 적응

드리프트로 인해 이동한 Base Logit을 샘플별로 보정하여, 모델의 드리프트 강건성 향상

3. 데이터 효율적 드리프트 대응

일반적인 Continual Learning 대비 적은 Target Year 샘플만으로 Drift 대응 가능

PART 02

Related Work

왜 기존 Android Malware Detector은 시간이 지나면 약해지는가?

Related Work

기존 연구가 어떻게 문제를 해결했는가

1. Continual Learning 기반

Chen et al. (USENIX Sec '23)

Active Learning + Contrastive Learning을 결합하여 재학습 샘플을 선별하고 드리프트 대응

LDCDroid (Computers & Security '25)

학습 데이터의 드리프트 특성을 학습하여 Active Learning + Pseudo-labeling으로 모델 노화 대응

2. Threshold, Score, Logit 조정 및 처리 기반

Roh et al. (KSC '25)

API 공출현 그래프의 구조 변화(Louvain 커뮤니티)를 이용해 드리프트를 정량화하고 탐지 임계값 조정

Raza et al. / Papadopoulos et al.

분류 Score를 확률적으로 보정하거나, 예측 Confidence Guarantee를 제공하여 불확실 샘플 보수적 처리

Transcend (USENIX Sec '17) / Guo et al. (ICML '17)

Logit 기반 Confidence를 믿을 수 없으니 재학습에서 제외 / 모든 샘플의 Logit을 전역 스칼라로 보정

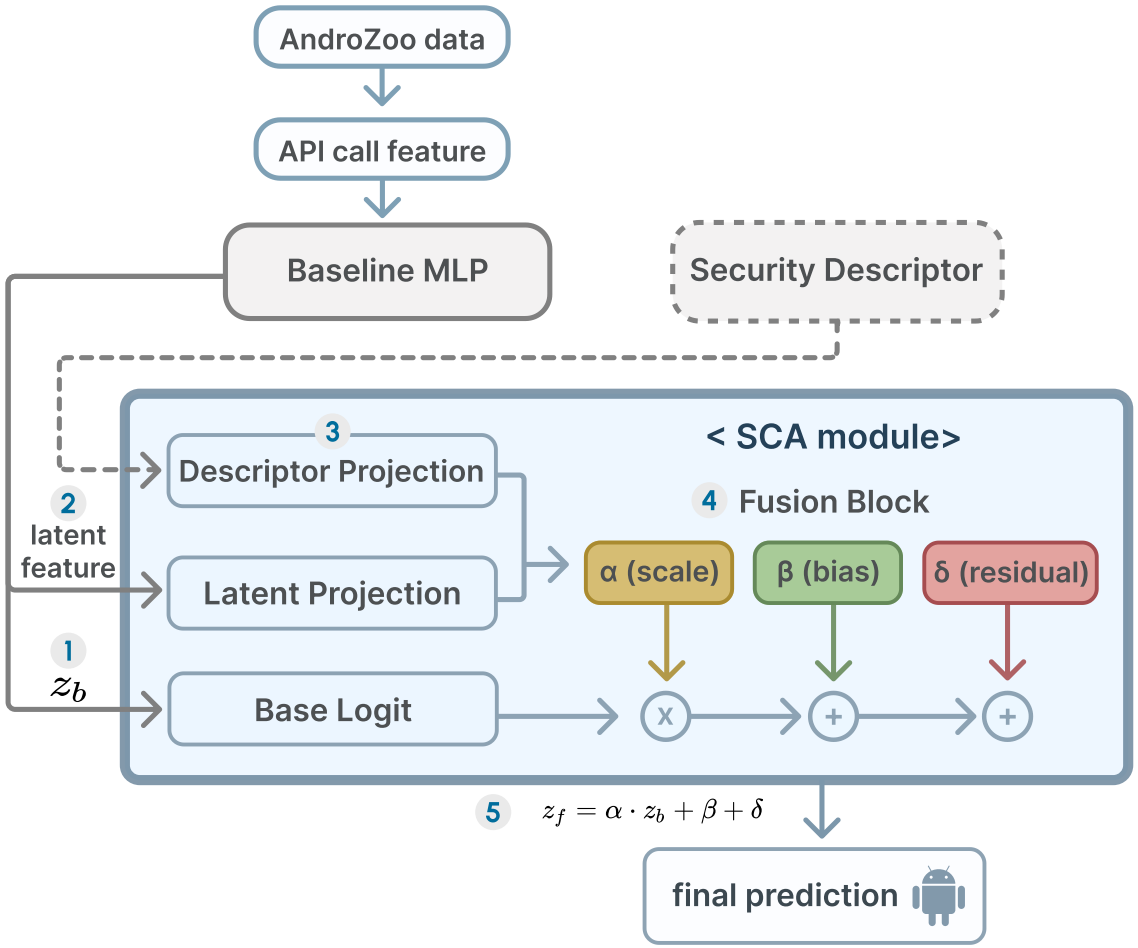
PART 03

Proposed Module

SCA: Security-Calibrated Adapter for Concept Drift in Android Malware Detection

SCA: Security-Calibrated Adapter 전체 구조

Concept Drift 환경에서 강건한 탐지 성능 유지를 위한 Logit 보정 모듈



Baseline MLP

2014~2018 데이터로 모델 학습 후
Base Logit z_b 생성

Concept Drift 발생 시

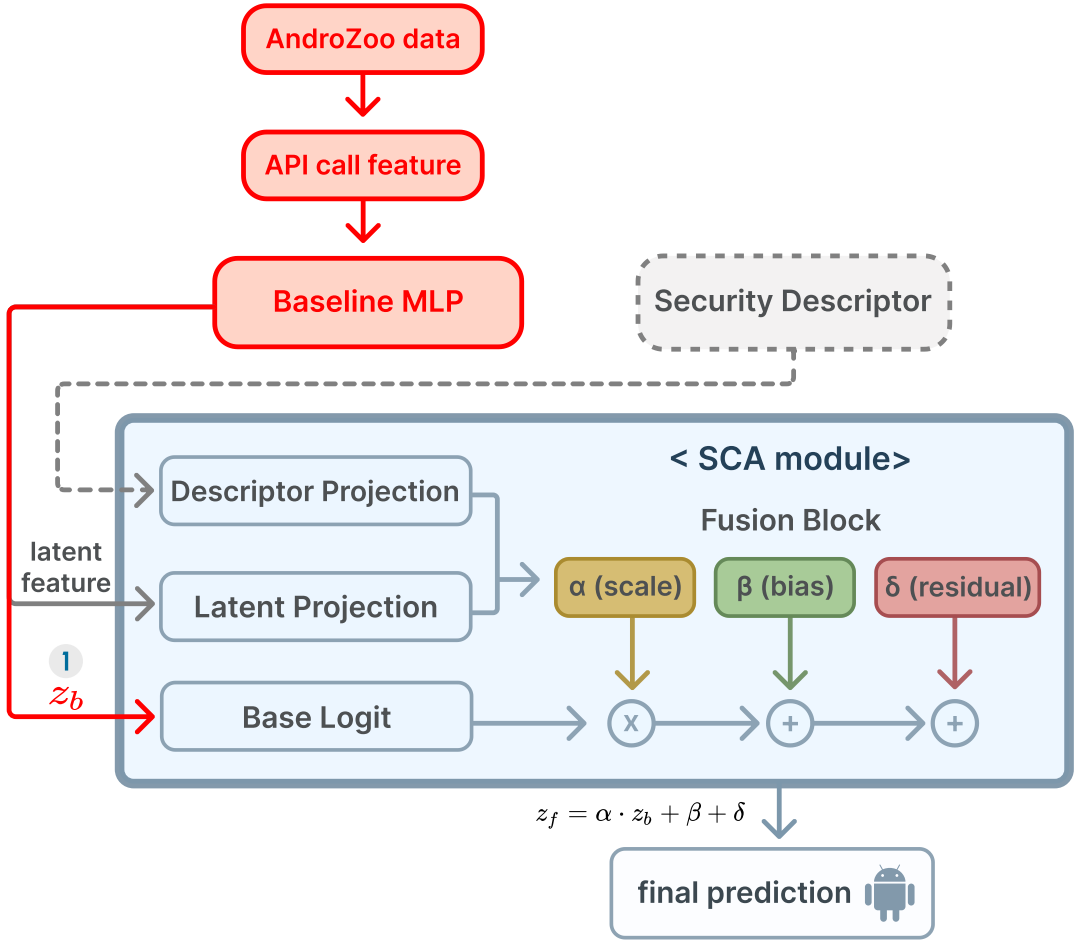
Base Logit shift로 인한 신뢰도 저하 발생

Baseline MLP 고정 & SCA 기반 적응

적응 비율에 따라 Target 연도 일부 라벨 샘플을 활용하여, SCA를 통해 Base Logit을 보정하고 Baseline 예측을 Target 환경에 적응

[Step 1] Baseline Classifier

API-call 기반 정형 feature 분류에 적합한 MLP를 Baseline Classifier로 사용

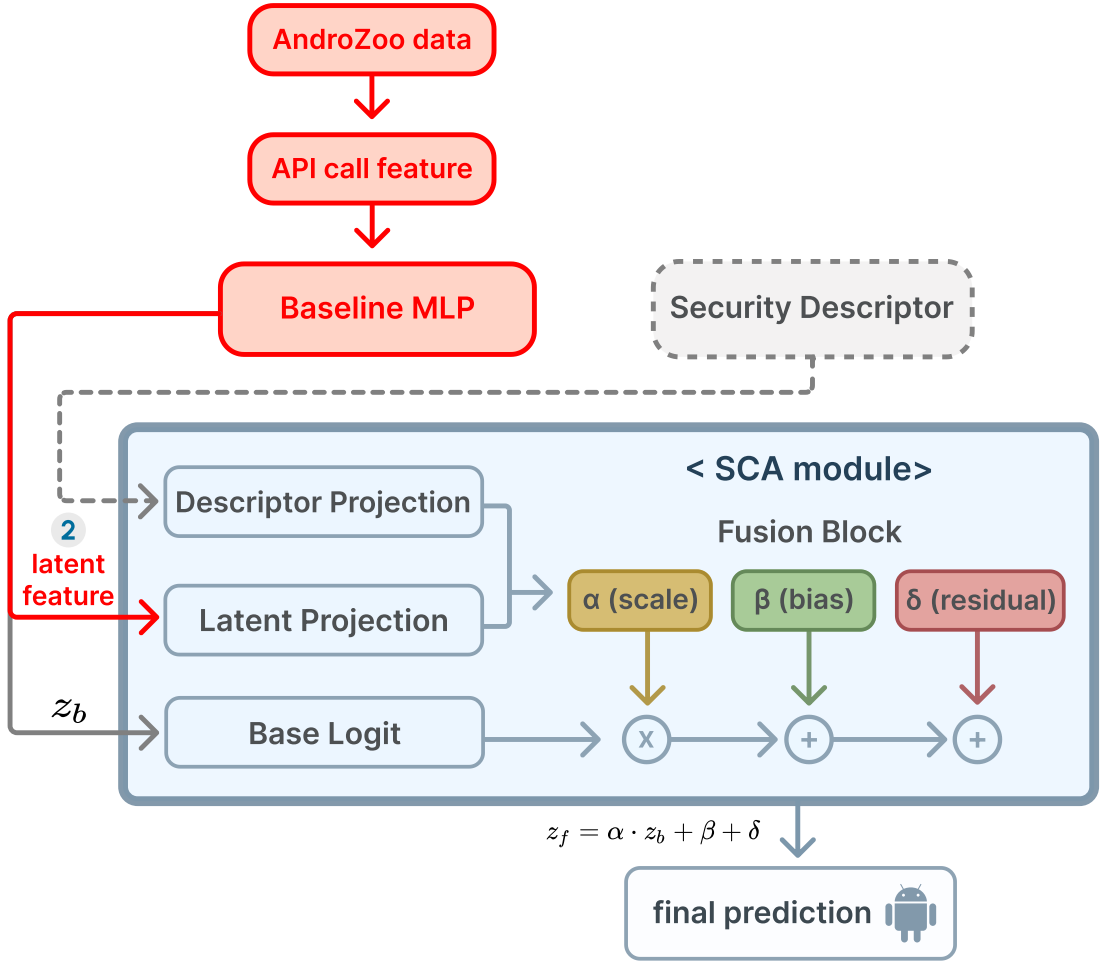


Base Logit

- 입력 API call feature를 기반으로 Base Logit z_b 생성
- **Baseline MLP가 해당 샘플을 얼마나 악성/정상으로 판단했는지** 나타내는 최종 판단값

[Step 2] Latent Feature

Base Logit 뿐만 아니라 Baseline 내부 표현도 함께 활용

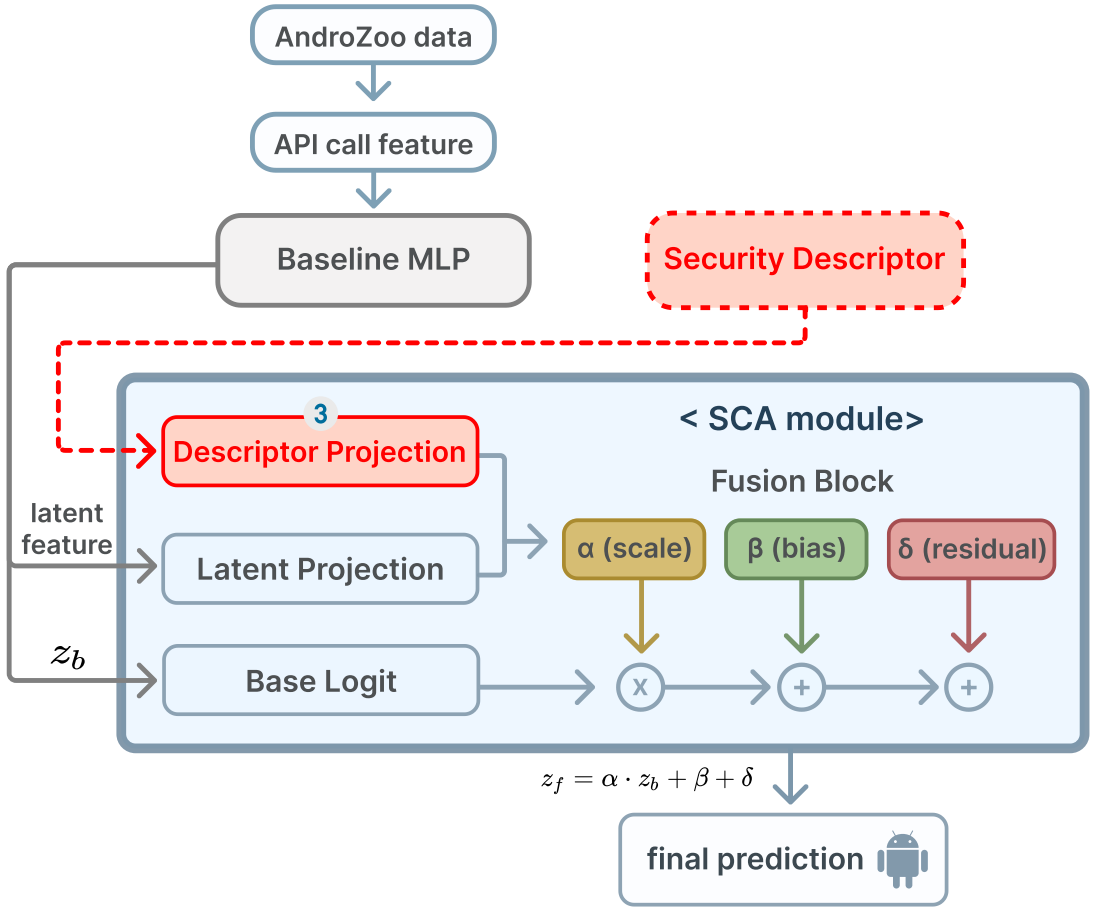


Latent Feature

- Baseline MLP의 **마지막 은닉층에서 추출한 고차원 특성**
- 최종 Logit으로 압축되기 전의 **특징 조합 및 패턴 정보 제공**
- 이후 Latent Projection을 통해 SCA module의 입력으로 사용되는 **Latent Embedding으로 변환**

[Step 3] Security Descriptor

보안 의미론적 특성을 반영한 도메인 지식 기반 표현



① 보안 행위 그룹 사전 정의 및 API call feature 매핑

분석가가 Android 악성 행위 의미를 기준으로 보안 행위 그룹과 API call feature를 룰 기반으로 매핑하여 사전 정의

② 입력 앱의 보안 행위 그룹별 특성 추출

입력 앱의 API Call Feature 기반으로 사전 정의된 보안 행위 그룹별 특성 계산

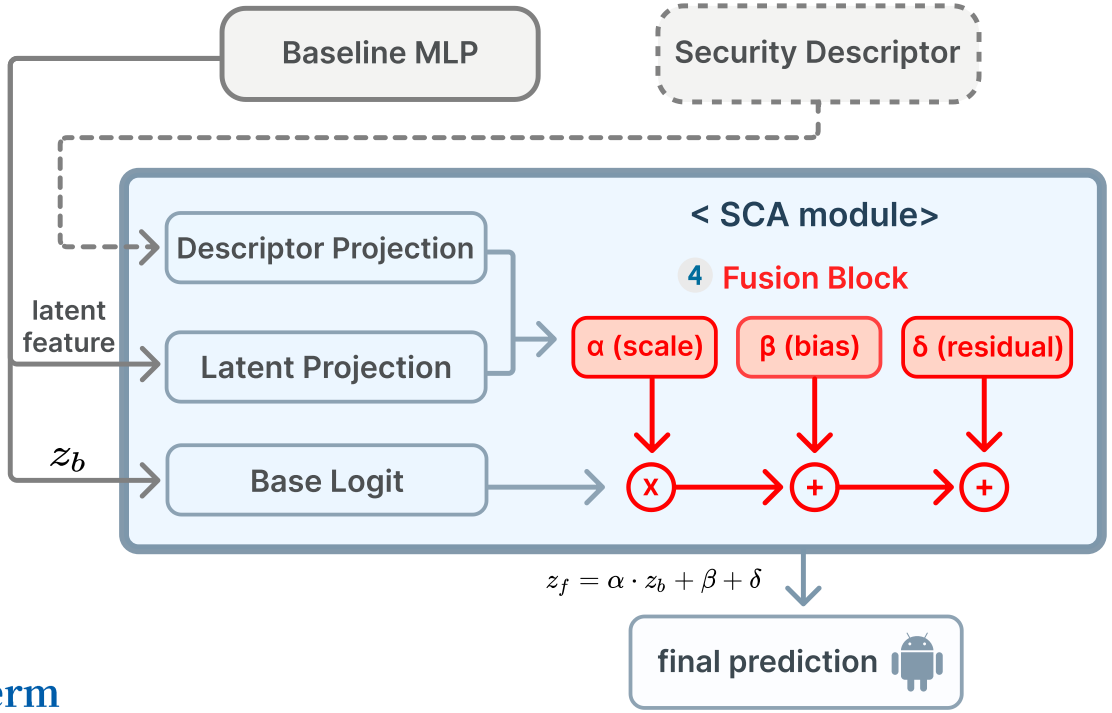
- 해당 보안 그룹 API의 존재 여부
- 그룹 내 API call frequency의 로그 합
- 그룹별 위험도 가중치 기반 점수
- 일부 보안 그룹 간 동시 출현 정보

③ Security Descriptor 생성 및 Embedding

보안 행위 그룹별 특성을 결합하여 Security Descriptor 벡터를 생성하고, Projection을 통해 SCA 입력용 Descriptor Embedding으로 변환

[Step 4] Fusion Block

Latent Feature, Base Logit, Security Descriptor를 입력으로 받아 각 샘플별 α, β, δ 생성



α: Scaling Term
Base Logit의 영향력을 샘플별로 조절하는 계수

β: Bias Term
Drift로 인해 발생한 Baseline의 전반적인 판단 편향 보정

δ: Residual Term
개별 샘플 특징 패턴을 반영한 추가 보정값

Base Logit
기존 Baseline MLP의 판단 강도를 제공하여, 기존 모델의 판단을 얼마나 반영하고 보정할지 결정하는 데 활용

Latent Feature
Baseline MLP가 학습한 샘플별 내부 패턴을 제공하여, 샘플 특성에 따른 적응적 보정 정보를 생성하는 데 활용

Security Descriptor
Android 보안 의미 정보를 제공하여, 보안 행위 패턴을 반영한 적응적 보정 정보를 생성하는 데 활용

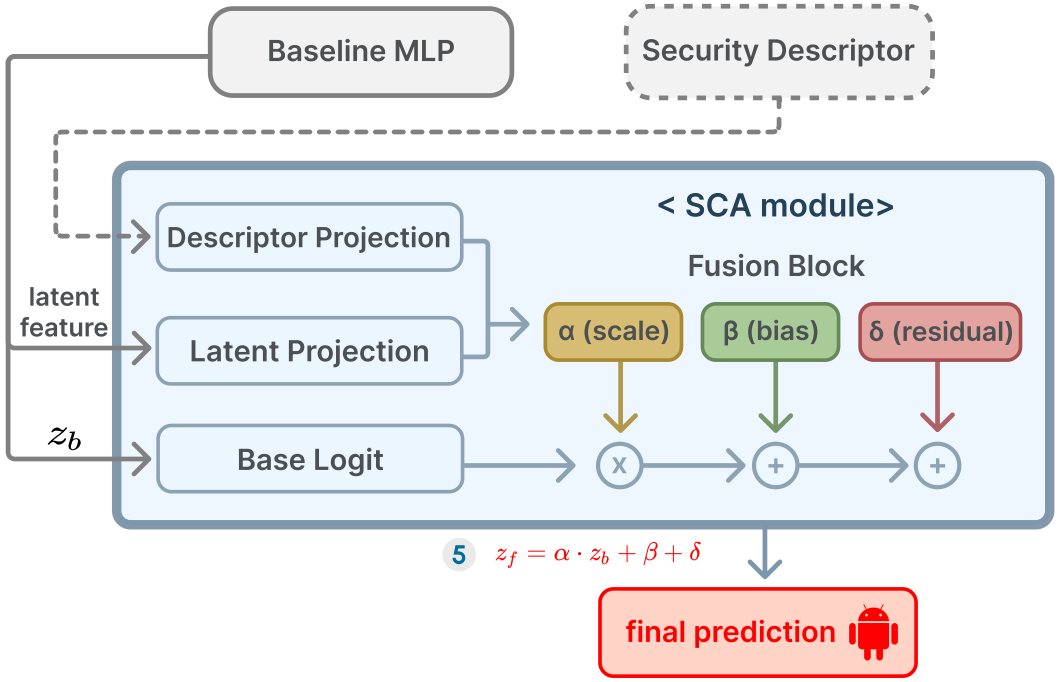
→ **세 입력은 Fusion Block에서 결합되어 α, β, δ 를 공동 생성한다.**

[Step 5] Logit Calibration

Step 4에서 생성된 α, β, δ를 수식에 적용하여 Base Logit을 최종 Logit으로 보정

$$z_f = \alpha \cdot z_b + \beta + \delta$$

z_f : 최종 보정 결과, z_b : Baseline 출력 기본 Logit



SCA Logit Calibration 전·후 Logit 비교

케이스	실제 라벨	logit	예측	결과
Baseline MLP	Benign	+5.65	Malware	Wrong
SCA	Benign	-4.77	Benign	Correct

→ 모델의 최종 예측은 기존 Base Logit이 아닌,
SCA가 보정한 Final Logit을 활용하여 드리프트 강건성 보장

PART 04

Evaluation

SCA는 Concept Drift 환경에서도 안정적인 탐지 성능을 유지하는가?

실험 환경 구축

데이터셋 구성 및 특징정보 추출

Dataset

- AndroZoo 기반 Android Malware Dataset 사용
- API Call Frequency Feature 사용
  (DEX File에서 추출)
- Concept Drift 평가를 위해 연도별 데이터 분리
- Train Set: 2014 ~ 2018
- Test Set: 2019 ~ 2023

Training Set			Test Set		
Year	Benign	Malicious	Year	Benign	Malicious
2014	5,000	5,000	2019	3,000	3,000
2015	5,000	5,000	2020	3,000	3,000
2016	5,000	5,000	2021	3,000	3,000
2017	5,000	5,000	2022	3,000	3,000
2018	3,000	3,000	2023	3,000	3,000
Total	23,000	23,000	Total	15,000	15,000

❖ 각 test year에서 적응 비율 만큼의 샘플을 SCA 기반 적응에 활용하고, 나머지 샘플만 최종 평가에 사용

실험 평가

비교 평가 대상 및 평가 지표 소개

제안 모듈과 비교군

Baseline MLP

별도의 적응을 수행하지 않는
기본 Baselline 모델이다.

Additive Adapter

제안 수식에서 **α를 고려하지 않고, β와 δ 기반의 가산형 보정만 적용**한다.

SCA (제안 모듈)

α, β, δ를 모두 적용하여 Base Logit을 샘플별로 보정한다.

평가 지표

Accuracy

전체 데이터 중 모델이 올바르게
분류하거나 예측한 비율

F1-score

정확도에서 데이터 불균형의 왜곡을
방지하기 위해 정밀도와 재현율의
조화평균을 계산한 것

실험 결과 - 평균 성능 비교

평가 데이터셋에 대한 평균 성능 비교

평가 데이터셋 평균 성능 비교 (2019-2023, %)

방법	F1-score	Accuracy
Baseline MLP	71.6	58.7
Additive Adapter	74.4	64.6
SCA (Proposed)	**91.8**	**91.5**

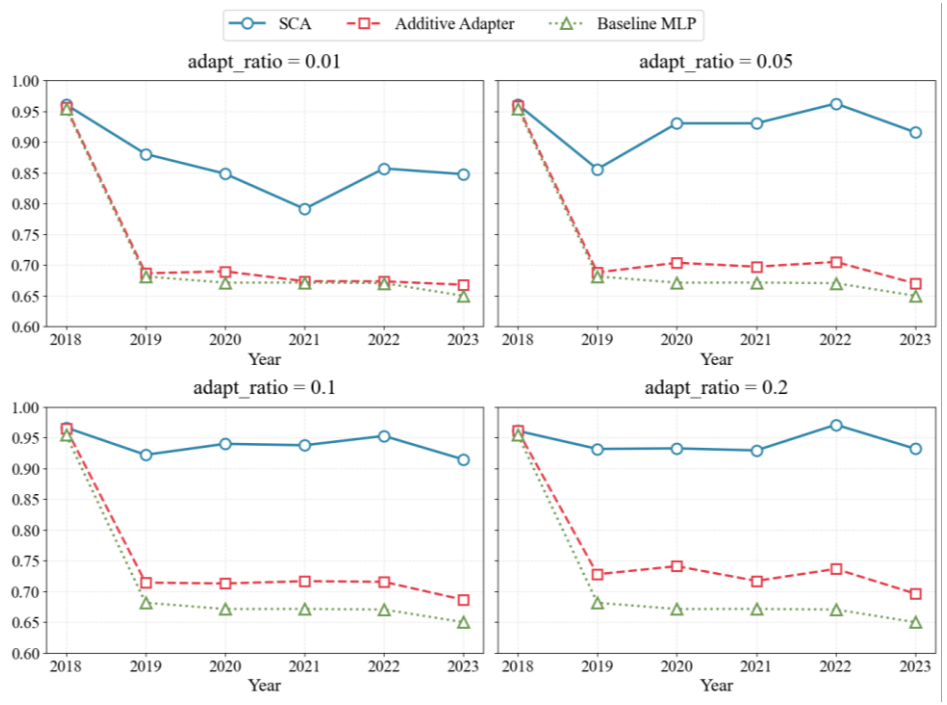
해석

- Baseline MLP는 적응 과정이 없어 모든 적응 비율에서 동일한 성능에 머묾.
- Additive Adapter는 일부 개선되지만, scaling term α의 부재로 드리프트 강건성이 떨어짐.
- **SCA**는 2019년 이후의 드리프트 구간에서도 **높은 탐지 성능을 안정적으로 유지**

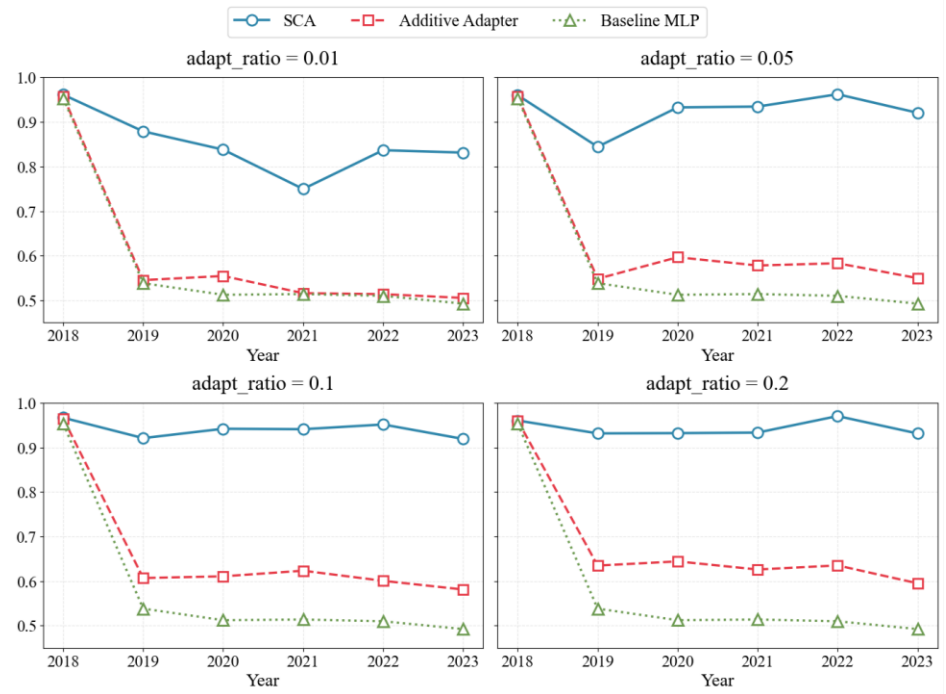
실험 결과 - 연도별 성능 추세

적응 비율 변화에도, SCA가 가장 안정적인 성능 추세를 보임

F1-Score



Accuracy



Baseline MLP - 급락 / 낮은 수준 유지

적응 메커니즘이 없어 **2019년 이후 성능이 크게 저하**

Additive Adapter - 개선되나 회복폭 제한

2019년 급락 후 **일부 성능 향상을 보이지만,** SCA 대비 명확한 성능 격차가 발생

SCA - 90% 이상 안정 유지

대부분의 적응 비율에서 2019년 이후에도 **높은 F1-Score와 Accuracy를 안정적으로 유지**

기존 연구와의 정성 비교

데이터·특징·드리프트 대응 관점에서 본 SCA의 차별점

구분	Chen et al.	LDCDroid	Proposed (SCA)
데이터 출처	AndroZoo	AndroZoo	**AndroZoo**
특징 정보	API Graph	API Graph	**API Call Frequency**
분류기	MLP + Encoder	MLP	**MLP**
드리프트 대응방식	Continual Learning	Continual Learning	**Logit Calibration**
연구 특징점	Active + Contrastive Learning	Active Learning + Pseudo-labeling	**Active + Adaptive Learning**
한계점	**Concept Drift 환경에서의 성능 격차 발생**	**높은 재학습 비용, 인간 전문가의 라벨링 필요**	**Descriptor가 사전 정의된 보안 그룹에 의존**

PART 05

Conclusion

우리 모듈의 기여와 한계 및 향후 연구

결론

SCA의 기여와 한계 및 후속 연구

결론

- Baseline 모델 재학습 없이 샘플별 Logit 보정을 통해 concept drift에 대응한다.
- Security Descriptor와 Latent Feature를 결합하여 단순 통계적 보정이 아닌, 보안 의미 기반 보정을 수행한다.
- 적응 비율을 통해 성능과 라벨링/적응 비용 사이의 균형을 운영 상황에 맞게 조절할 수 있다.

한계 및 후속 연구

- 완전한 unsupervised 방식은 아니며, adaptation 샘플이 필요
- Descriptor가 사전 정의된 보안 위험 그룹에 의존
- 향후 자동 descriptor 학습, unlabeled target data 활용, 다양한 baseline 모델 확장 필요

Q & A