정적 분석 알람을 위한 확률 모델 학습

김현수[1], Mukund Raghothaman[2], 허기홍[1]

[1]KAIST, [2]University of Southern California

@제주, KCC 2022

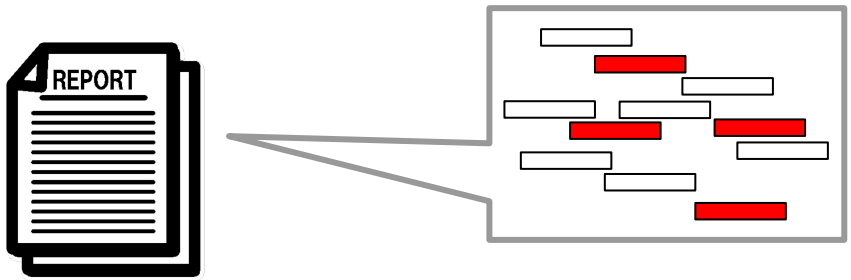
KAIST

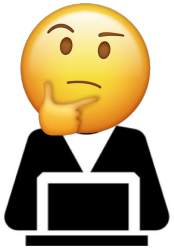
USC University of Southern California

정적 분석

프로그램 분석기 알람 보고서

정적 분석의 한계

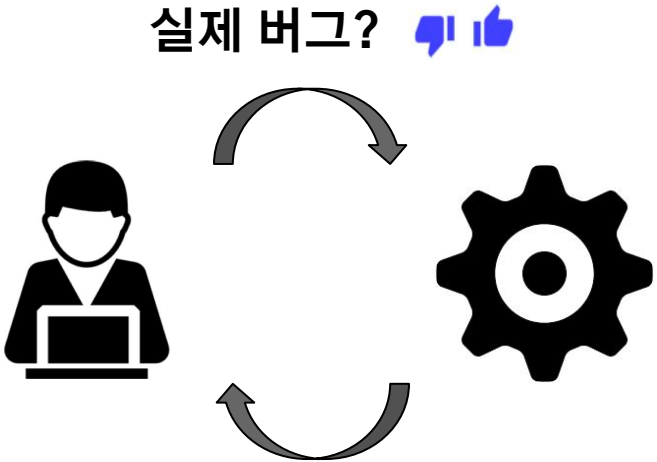


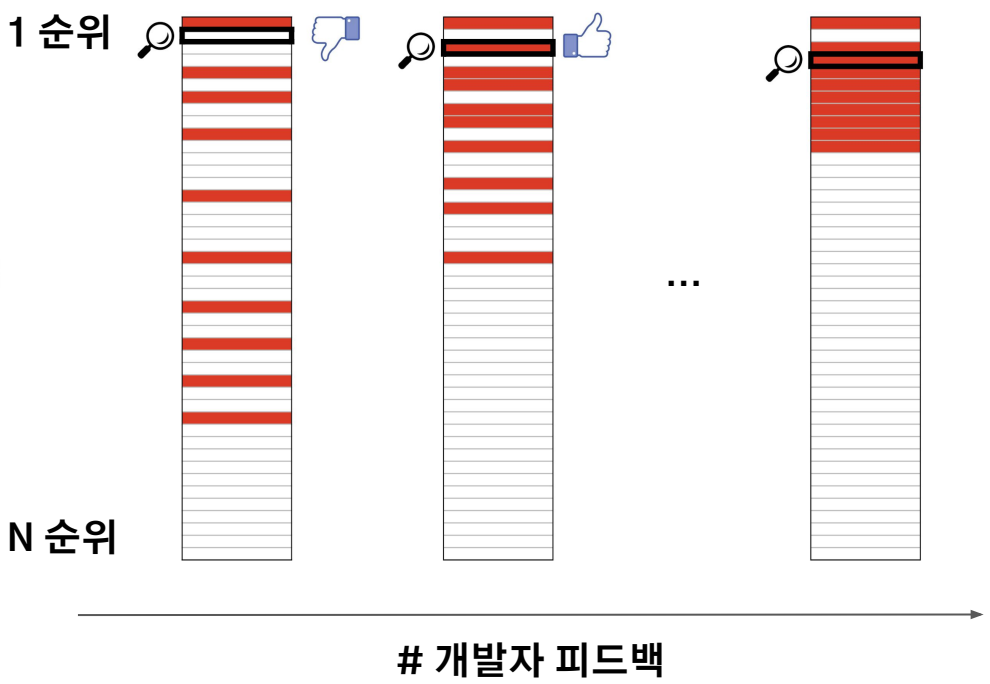


: 진짜 알람 (버그)

: 거짓 알람

베이지안 알람 랭킹 시스템 [PLDI'18, PLDI'19, FSE'21]





알람 랭킹 시스템의 성능

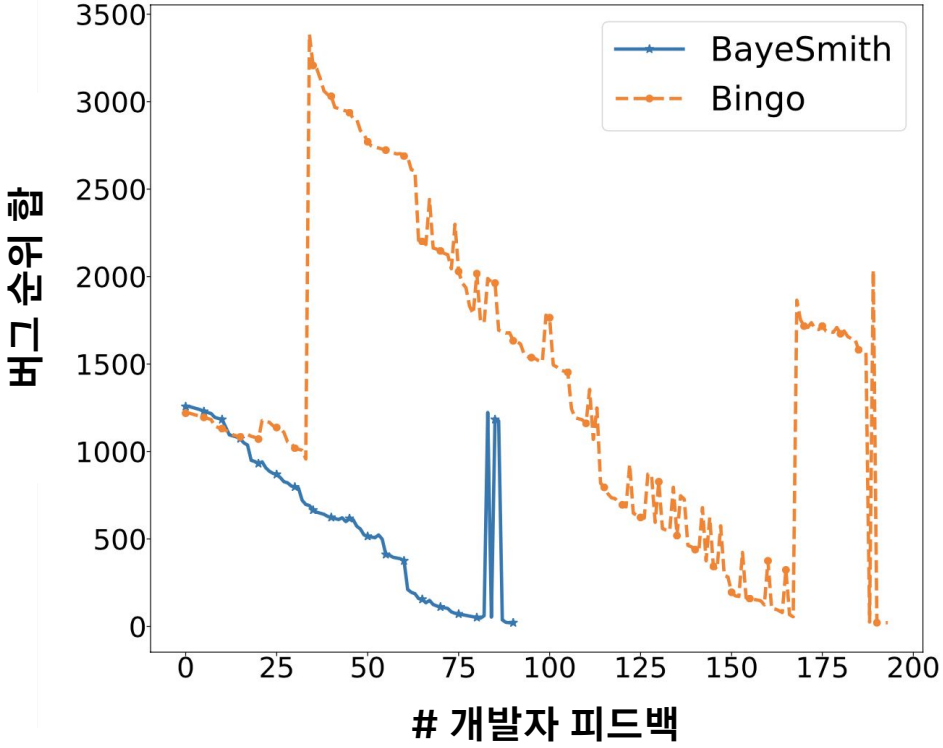
65 KLoC

wget

버퍼오버런 탐지를 위한
안전한 인터벌 분석
(sound interval analysis)

알람 : 891

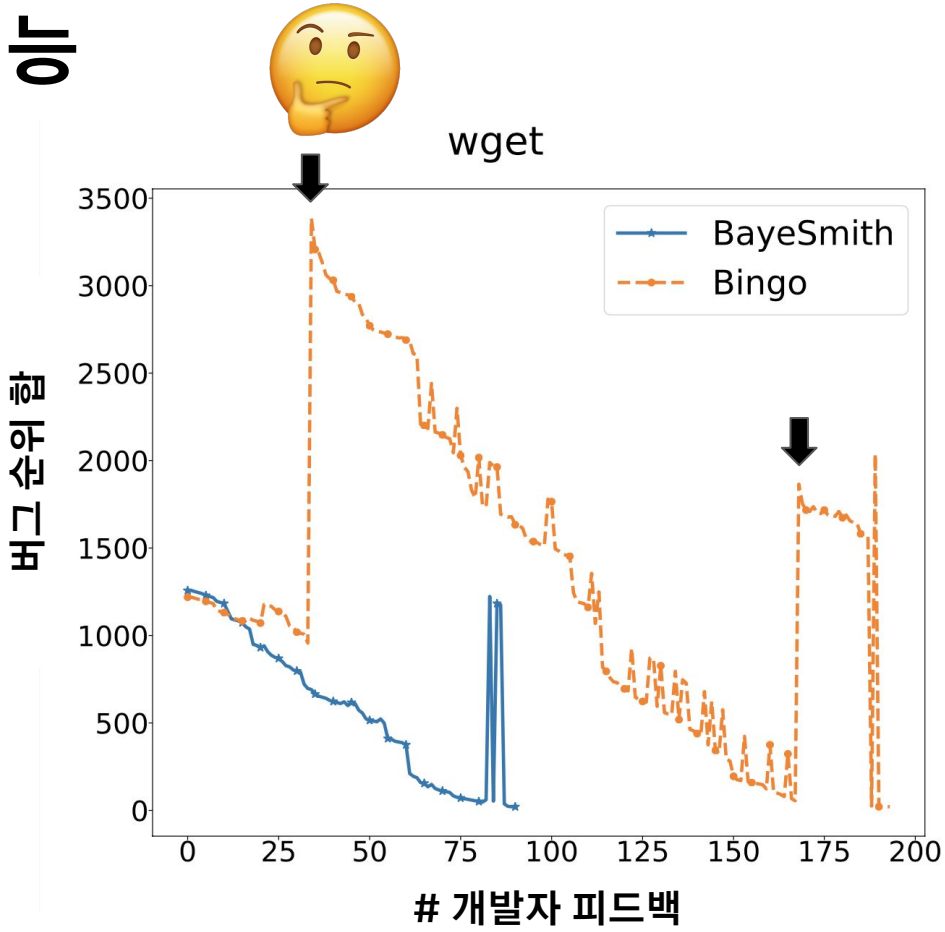
버그 : 6



알람 랭킹 시스템의 성능

⏰ # 알람 : 891

🐞 # 버그 : 6



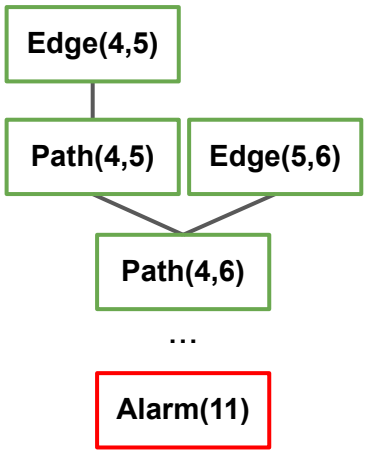
알람 랭킹 시스템 구성 과정

```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, 'r');
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, "_");
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;   // false alarm #1
9      p--;
10   }
11   if (*(p - 1) != "^")           // true alarm (buffer underflow)
12     *p = '\0';                   // false alarm #2
13 }
```

wget-1.2 일부

알람 랭킹 시스템 구성 과정

간략화한
인터벌 분석

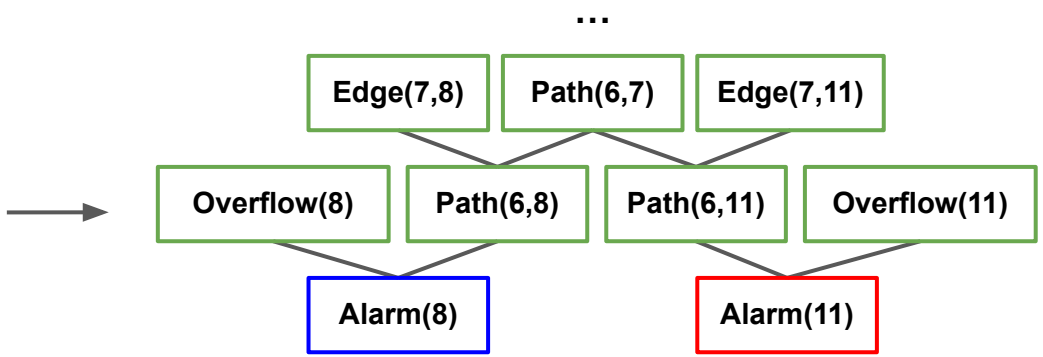



```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, 'r');
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, "_");
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;   // false alarm #1
9      p--;
10   }
11   if (*(p - 1) != "^")           // true alarm (buffer underflow)
12     *p = '\0';                   // false alarm #2
13 }
```

wget-1.2 일부

알람 랭킹 시스템 구성 과정

```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, ‘r’);
3    char *line, *tok;
4    line = read_line(fp);
5    tok = strtok(line, “_”);
6    char *p = tok + strlen(tok);
7    while (p > tok) {
8      if (!c_isdigit(*p)) break;
9      p--;
10   }
11   if (*(p - 1) != “^”)
12     *p = ‘\0’;
13 }
```



논리 표현
(정의–사용 관계)

알람 랭킹 시스템 구성 과정

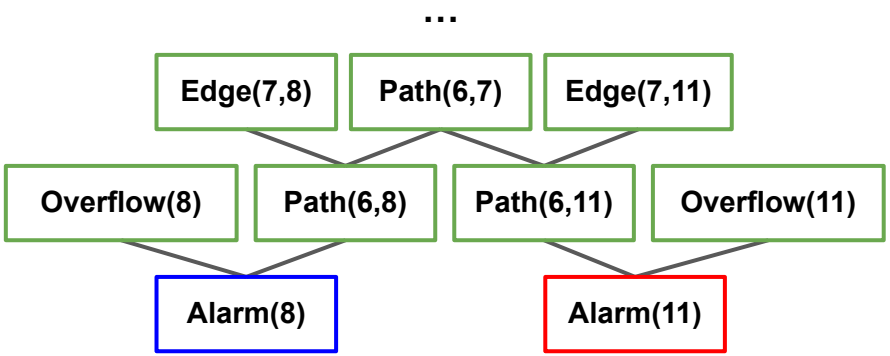
```
1  void ftp_parse_vms_ls (char *file) {
2    FILE *fp = fopen(file, 'r');
3    char *line, *tok;
4    line = read_line(fp);
5    tok
```

Path(x, y
Path(x, y) :- Path(x, z) /\ Edge(z, y)
Alarm(y) :- Path(x, y) /\ Overflow(y)

x에서 y까지 데이터 흐름,

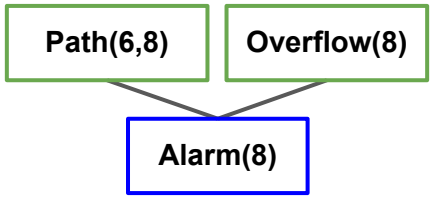
y에 오버플로우가 관측이 되면,

y에서 알람 도출



논리 표현
(정의-사용 관계)

베이지안 네트워크



Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y)
Alarm(y) :- Path(x, y) /\ Overflow(y)

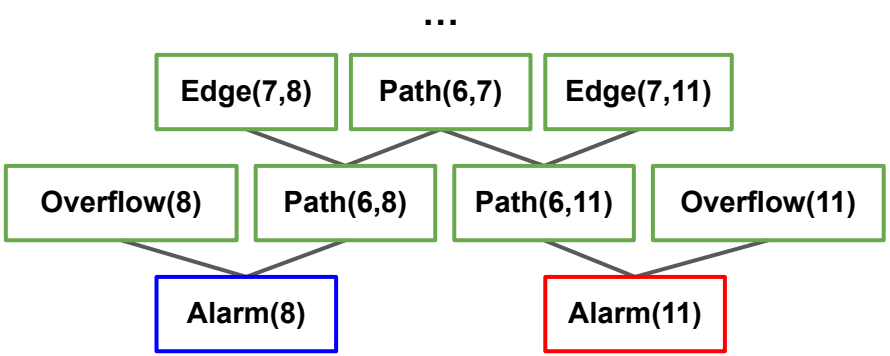
논리 규칙

Path(6,8)	Overflow(8)	Pr(Alarm(8) | H)
TRUE	**TRUE**	**0.99***
FALSE	TRUE	0
TRUE	FALSE	0
FALSE	FALSE	0

*학습 가능한 하이퍼파라미터.

확률 규칙

베이지안 네트워크 – 확률 추론

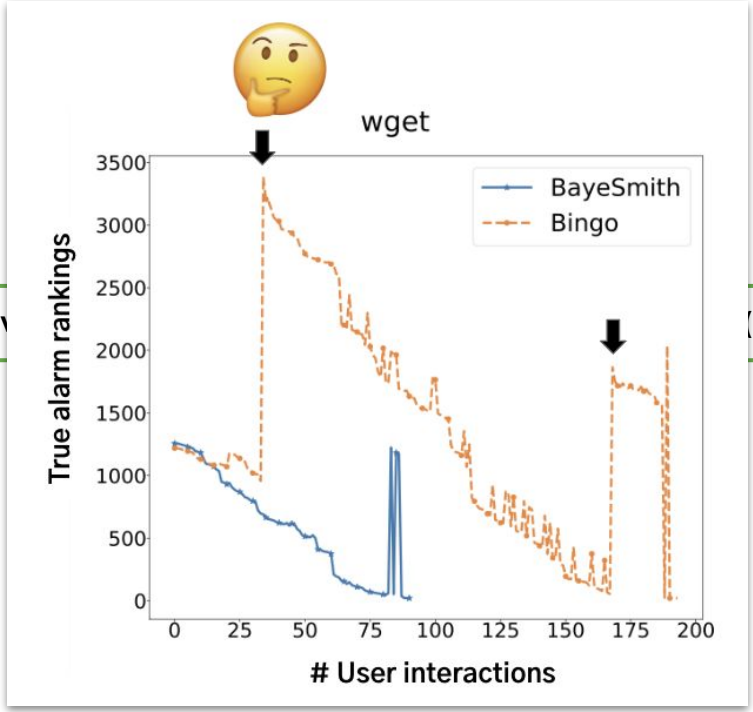


$$\Pr(\text{Alarm}(8)) = \Pr(\text{Alarm}(8) \mid \text{Path}(6,8), \text{Overflow}(8)) \times \Pr(\text{Path}(6,8) \mid \text{Path}(6,7), \text{Edge}(7,8)) \times \Pr(\text{Path}(6,7) \mid \ldots) \times \Pr(\text{Edge}(7,8)) \times \Pr(\text{Overflow}(8))$$

베이즈 법칙의 연쇄 적용

확률 표현
(베이지안 관계)

거짓 일반화 문제



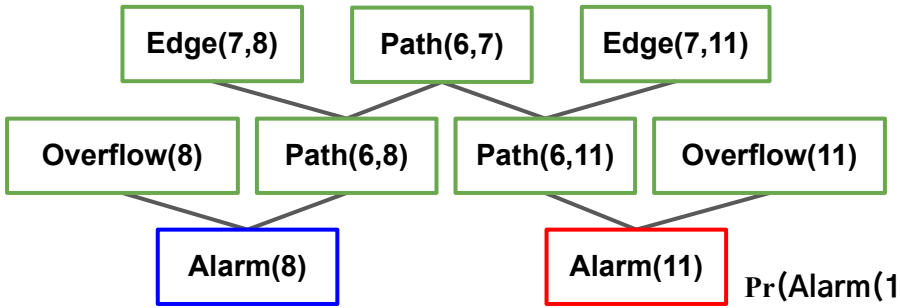
순위	알람	확률
1	Alarm(8)	**0.96**
2	Alarm(11)	**0.91**
...	...	...

➡

순위	알람	확률
...	...	...
136	Alarm(11)	**0.40**
...	...	...
-	Alarm(8)	-

거짓 일반화 문제

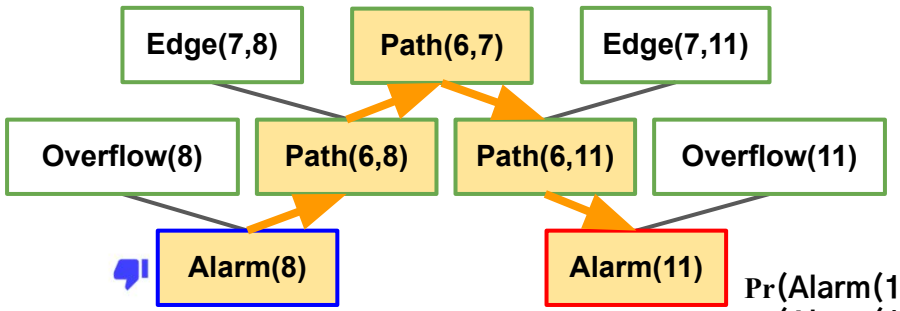
$\Pr(\text{Path}(6,7)) = 0.93$



$\Pr(\text{Alarm}(11)) = 0.91$

거짓 일반화 문제

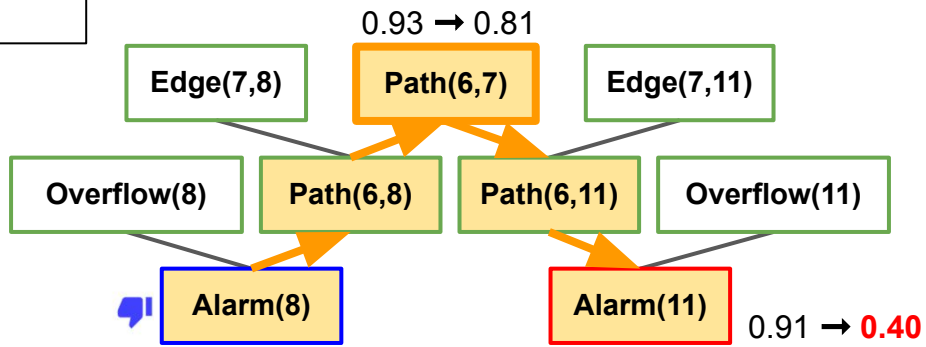
$\Pr(\text{Path}(6,7)) = 0.93$
$\Pr(\text{Path}(6,7) \mid \neg\text{Alarm}(8)) = 0.81$



$\Pr(\text{Alarm}(11)) = 0.91$
$\Pr(\text{Alarm}(11) \mid \neg\text{Alarm}(8)) = 0.40$

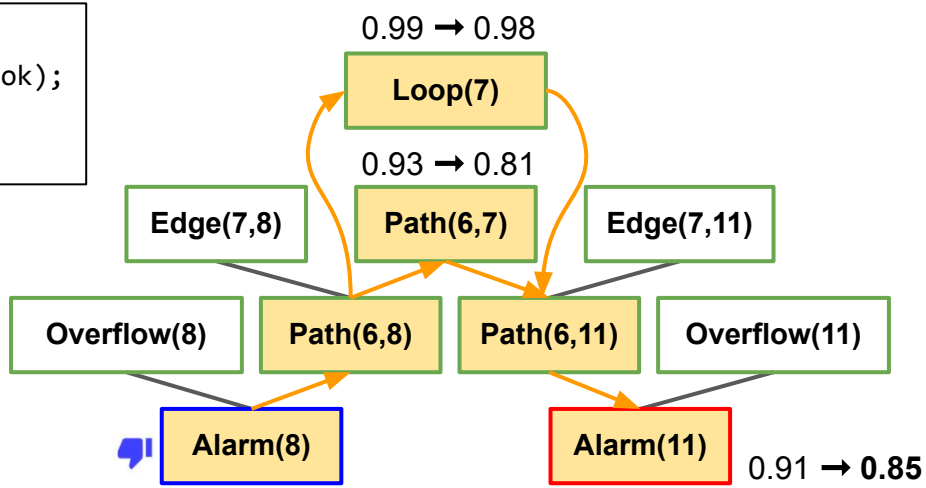
베이지안 알람 랭킹 시스템 개선 방안

```
5  ...
6  char *p = tok + strlen(tok);
7  while (p > tok) {
8    ...
```



베이지안 알람 랭킹 시스템 개선 방안

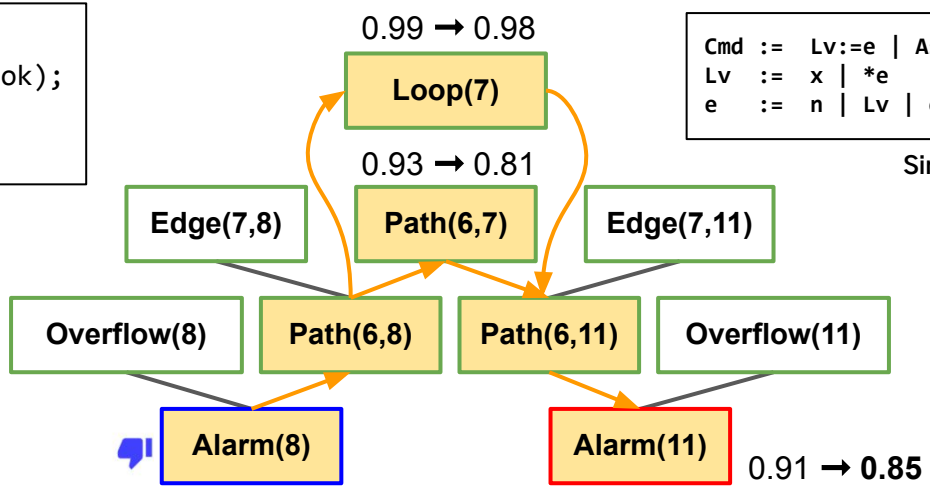
```
5  ...
6  char *p = tok + strlen(tok);
7  while (p > tok) {
8    ...
```



새로운 정보를 추가하여 오탐의 비난을 효과적으로 분산

베이지안 알람 랭킹 시스템 개선 방안

```
5  ...
6  char *p = tok + strlen(tok);
7  while (p > tok) {
8    ...
```

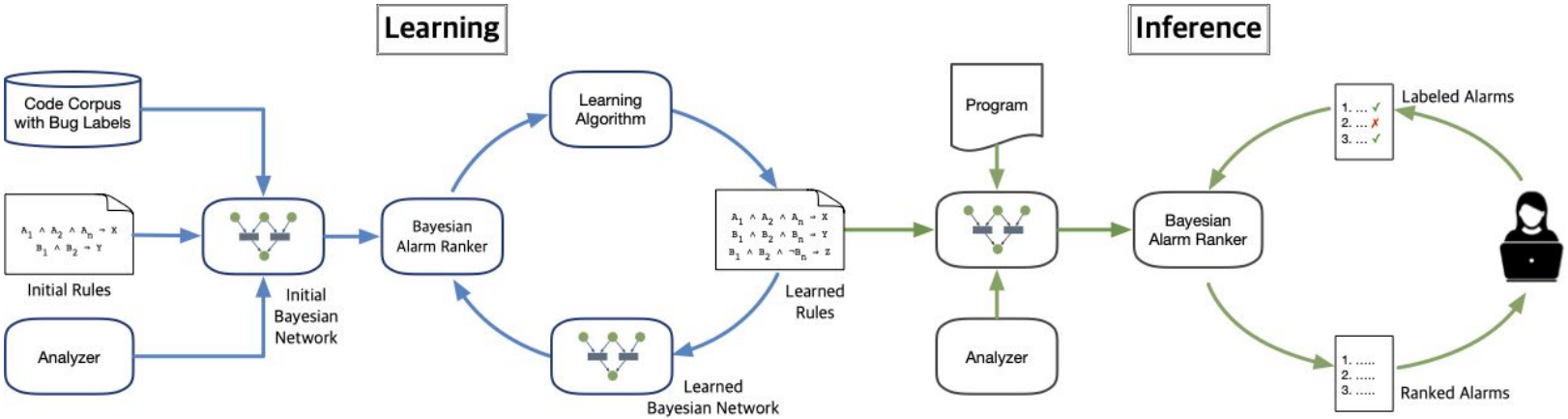



```
Cmd :=  Lv:=e | Assume(e) | Call(e,e) | Loop(e)
Lv  :=  x | *e
e   :=  n | Lv | e+e | ...
```

Simplified C syntax

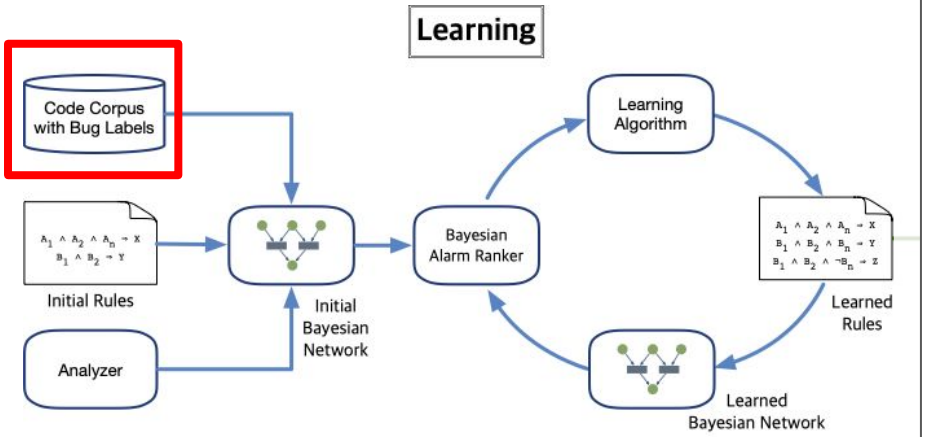
문법 정보를 추가하여 오탐의 비난을 효과적으로 분산

베이지안 알람 랭킹 시스템 학습 파이프라인



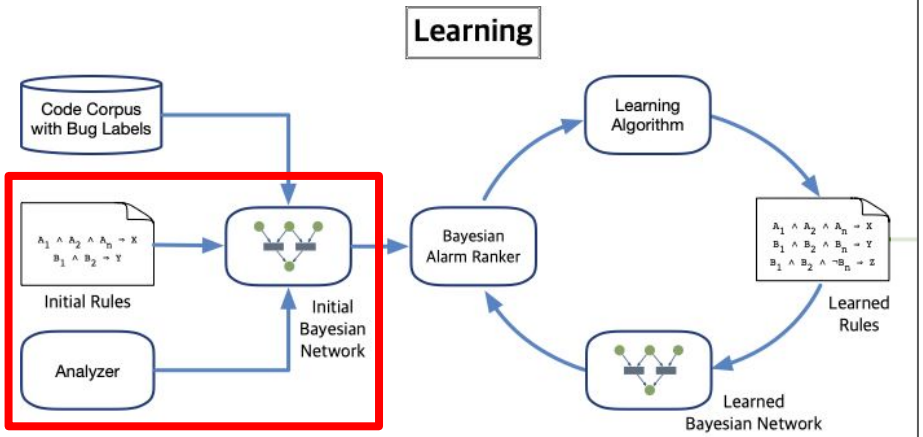
BayeSmith

베이지안 알람 랭킹 시스템 학

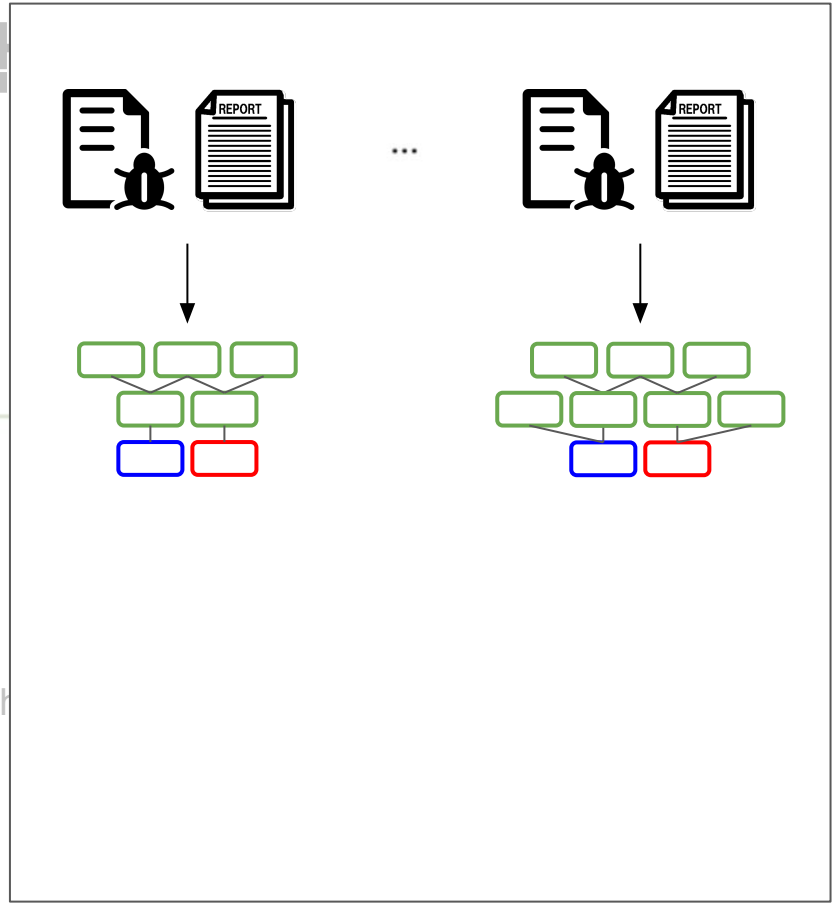


BayeSmith

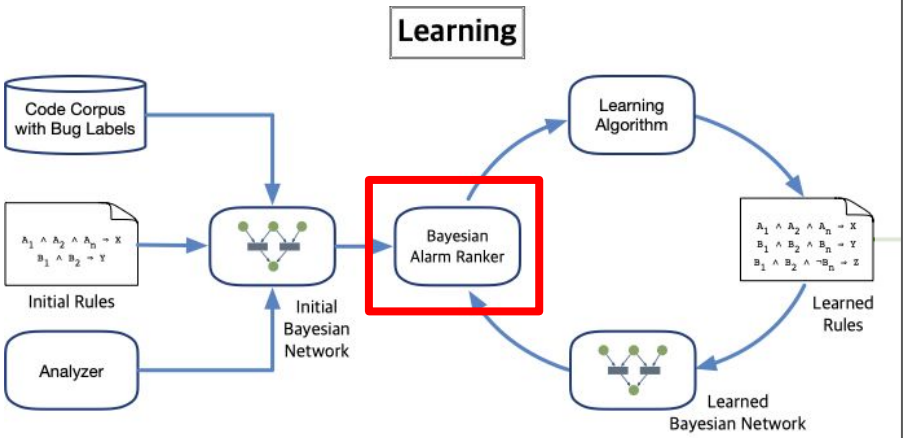
베이지안 알람 랭킹 시스템 학



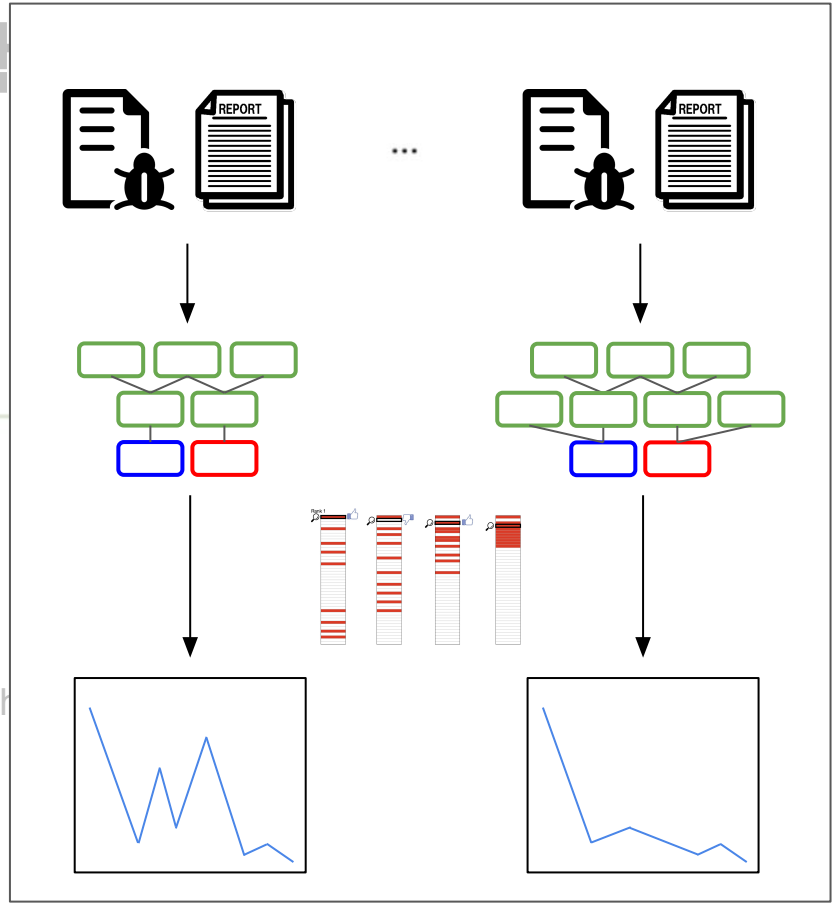
BayeSmith



베이지안 알람 랭킹 시스템 학

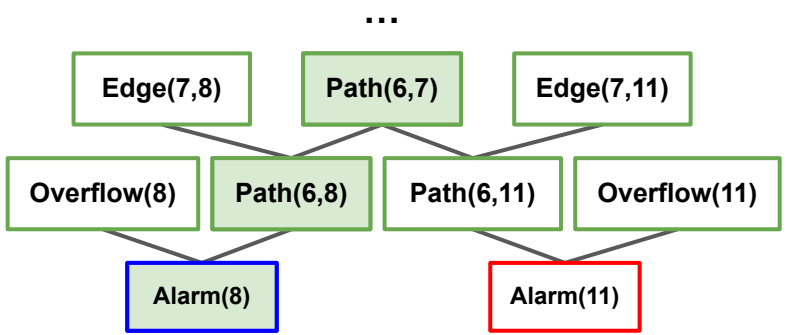


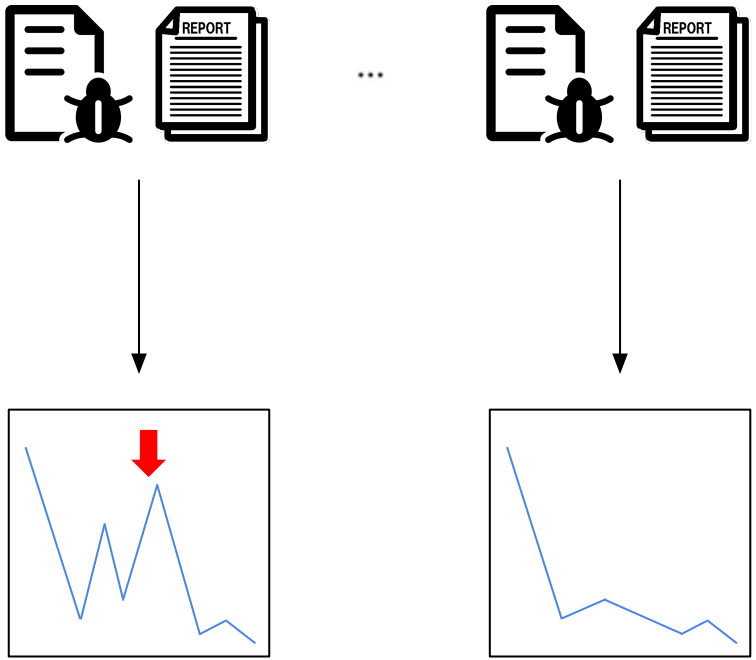
BayeSmith



베이지안 알람 랭킹 시스템 학습 파이프라인

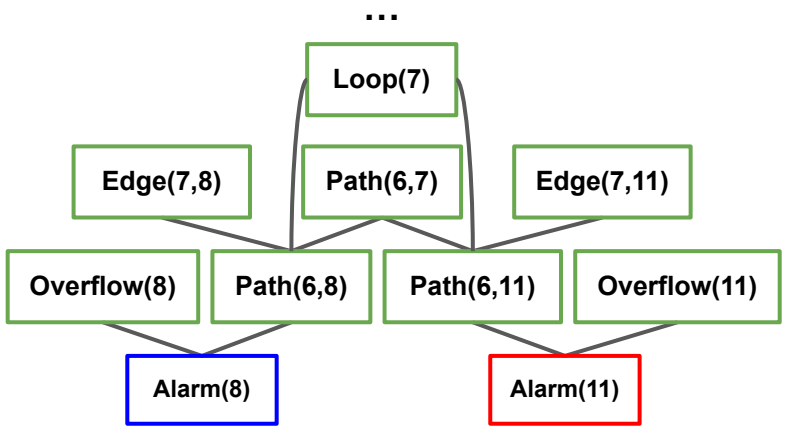
Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y)
Alarm(y) :- Path(x, y) /\ Overflow(y)

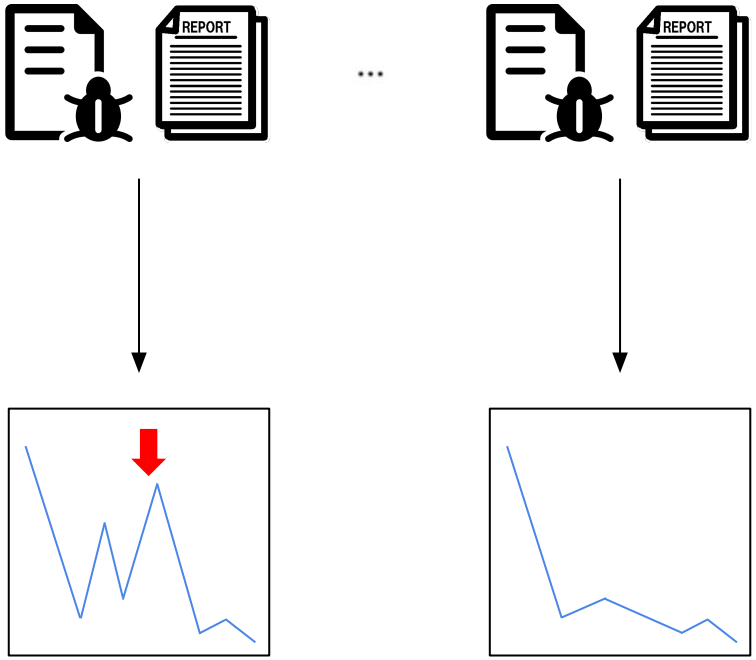




베이지안 알람 랭킹 시스템 학습 파이프라인

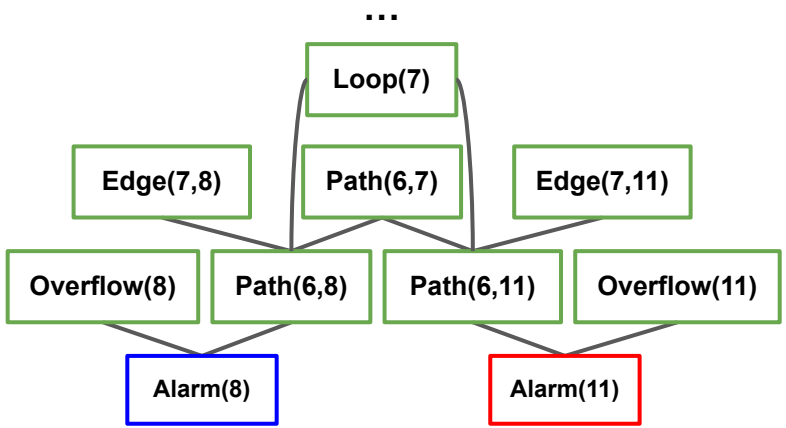
```
Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ Loop(z)   // p
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ !Loop(z)  // .99
Alarm(y)   :- Path(x, y) /\ Overflow(y)
```





베이지안 알람 랭킹 시스템 학습 파이프라인

```
Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ Loop(z)   // p
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ !Loop(z)  // .99
Alarm(y)   :- Path(x, y) /\ Overflow(y)
```

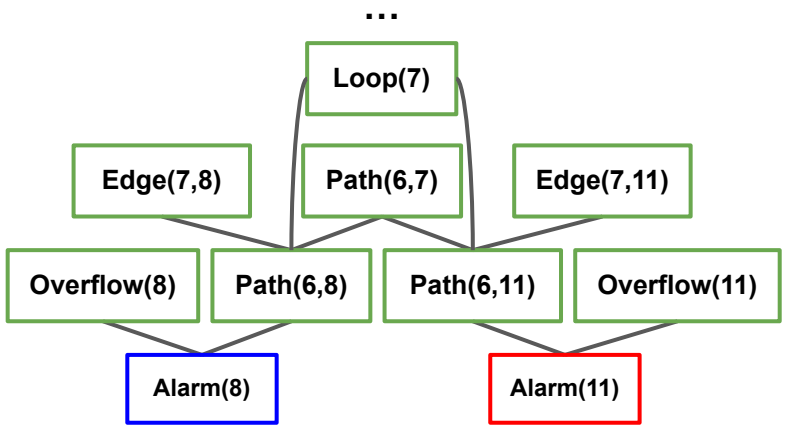


최소 50% 개선된 경우
=> 학습 결과 채택

Prog	Before	After
1	145	107
2	6	3
3	54	60
4	122	121

베이지안 알람 랭킹 시스템 학습 파이프라인

```
Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ Loop(z)   // p
Path(x, y) :- Path(x, z) /\ Edge(z, y) /\ !Loop(z)  // .99
Alarm(y)   :- Path(x, y) /\ Overflow(y)
```

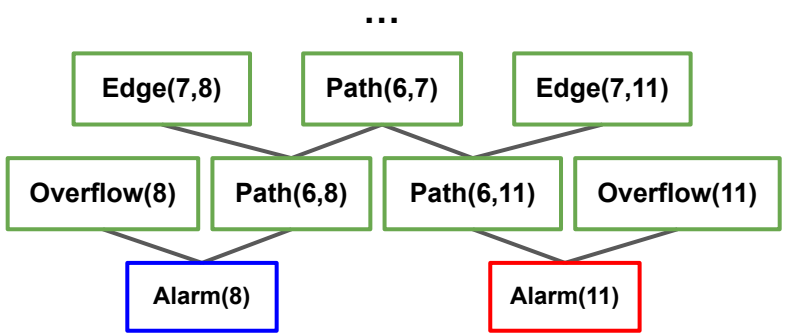


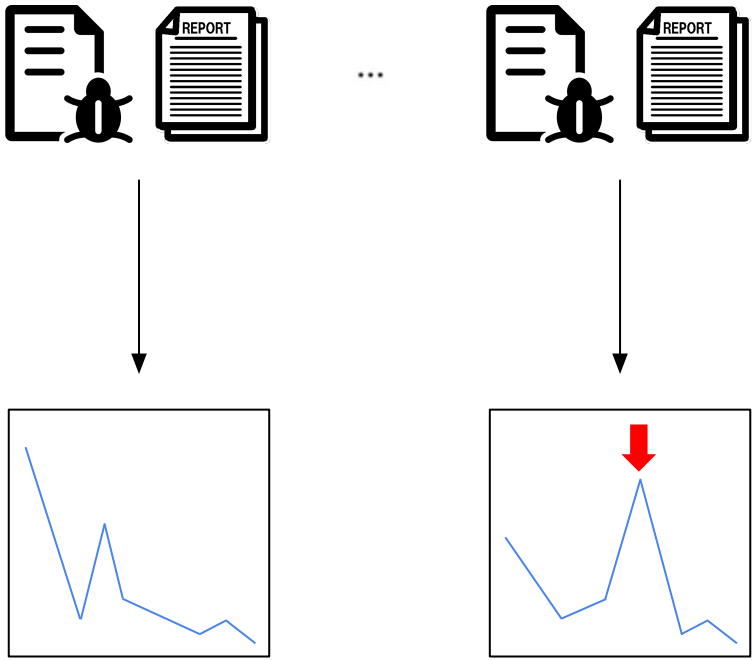
50%보다 적게 개선
=> 학습 결과 번복

Prog	Before	After
1	145	176
2	6	5
3	54	89
4	122	127

베이지안 알람 랭킹 시스템 학습 파이프라인

```
Path(x, y) :- Edge(x, y)
Path(x, y) :- Path(x, z) /\ Edge(z, y)
Alarm(y)   :- Path(x, y) /\ Overflow(y)
```





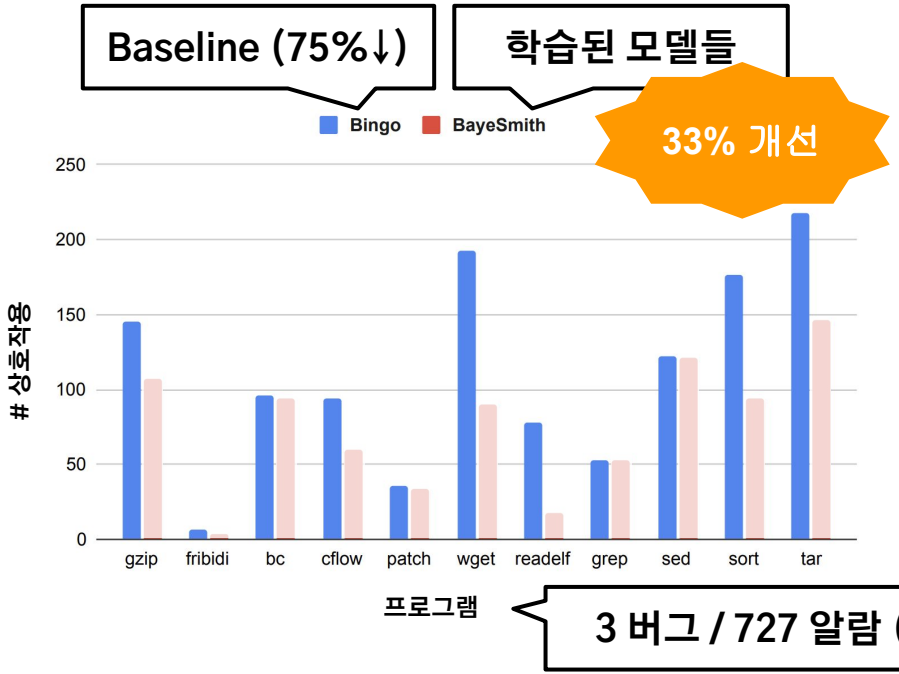
실험 방법

1. **벤치마크 구성**
   - 다양한 크기 (9~112 KLoC)의 GNU 프로그램들로 구성
   - 인터벌 분석 (11개), 테인트 분석 (9개)

2. **학습 방법**
   - 한 프로그램을 테스트 데이터, 나머지를 훈련 데이터

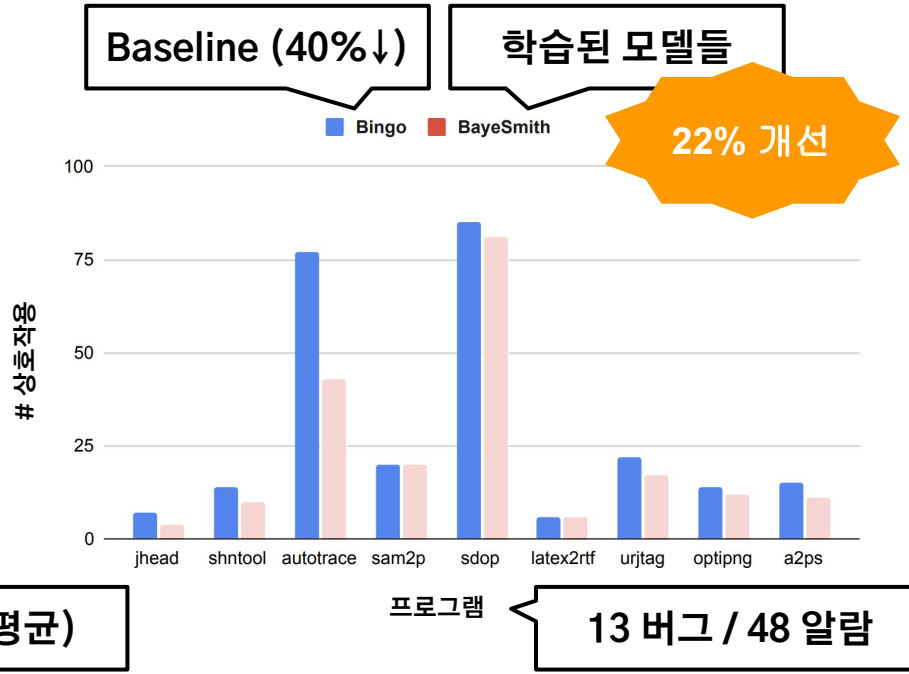
테스트 데이터 **훈련 데이터**

실험 결과 – 인터벌 & 테인트 분석



3 버그 / 727 알람 (평균)

인터벌 분석
(버퍼 오버런)



13 버그 / 48 알람

테인트 분석
(정수 오버플로우, 형식 문자열)

결론

- 정적 분석 알람을 위한 확률 모델의 일반적인 학습 프레임워크.
- 학습된 모델들의 효과적인 거짓 일반화 문제 완화를 체계적인 실험을 통해 검증.

감사합니다