Xinlong PAN, Jianhua LI, Zhihong ZHOU, Gaolei LI, Xiuzhen CHEN, Jin MA, Jun WU, Quanhai ZHANG, 2025. Large language model-enhanced probabilistic modeling for effective static analysis alarms. *Frontiers of Information Technology & Electronic Engineering*, 26(10):1926-1941. https://doi.org/10.1631/FITEE.2500038

Large language model-enhanced probabilistic modeling for effective static analysis alarms

Key words: Static analysis; Bayesian inference; LLM; Alarm ranking

Xinlong PAN
E-mail: mr.p332@sjtu.edu.cn
ORCID: https://orcid.org/0009-0006-3328-4080

Motivation

1. Traditional static analysis alarm prioritization methods suffer from notable drawbacks: probability models **relying on user feedback are limited by manually designed rules**, prone to false generalization, and lack intelligent optimization; meanwhile, **learning-based approaches face high training costs and constraints from predefined model structures**, failing to effectively reduce the burden of manual alarm verification for users.

2. The integration of large language models (LLMs) into static analysis has not yet realized its full potential—LLMs still **suffer from issues like hallucinations and randomness**, leading to limited vulnerability identification accuracy. Moreover, most existing methods use LLMs only as judgment substitutes or simple supplements, **lacking deep integration with static analysis, and are restricted by data processing granularity (e.g., function or line-level)**, making it difficult to address scenarios such as cross-library vulnerability detection.

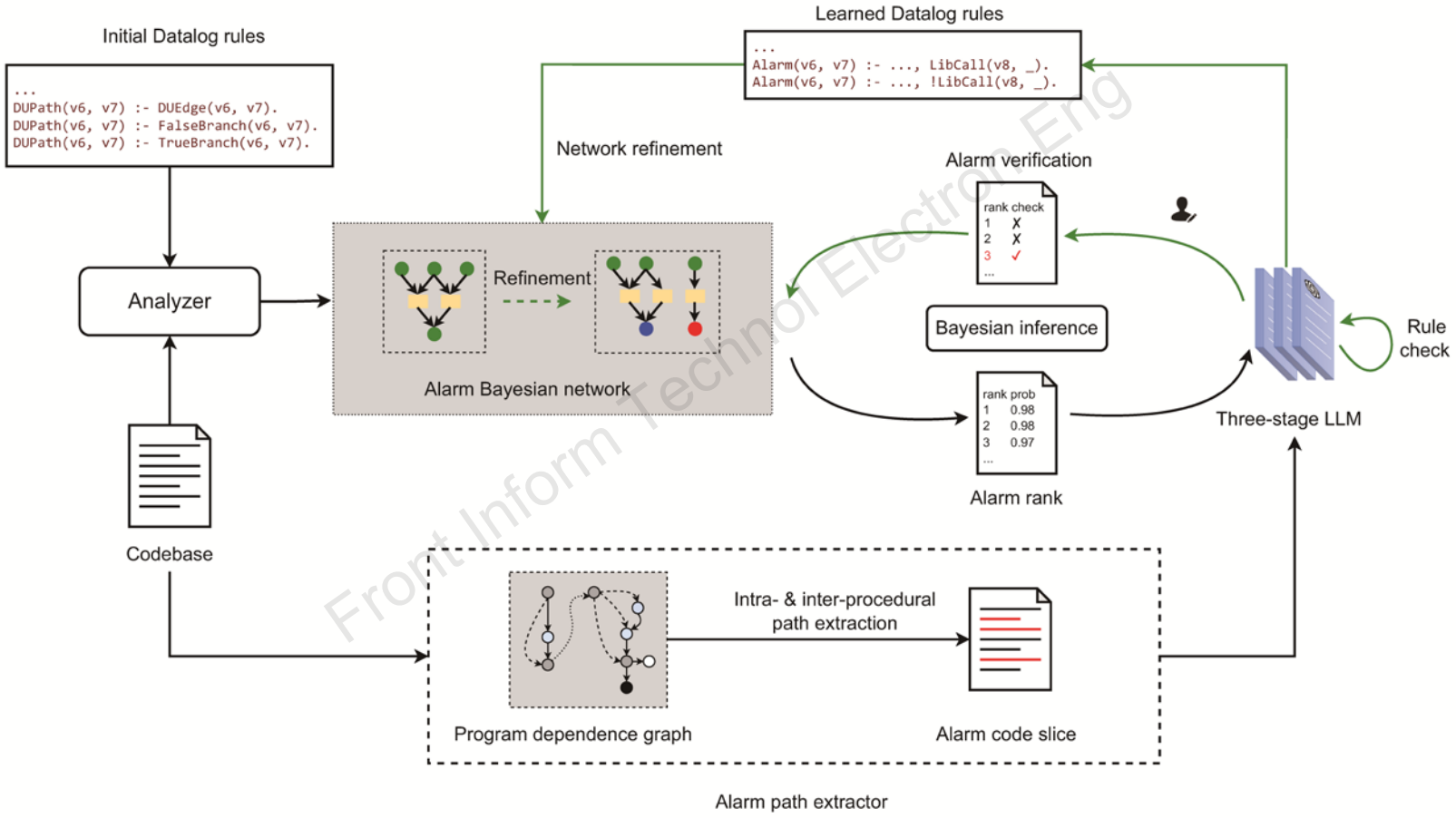
Main idea

1. **Propose the BinLLM framework**: Integrate LLMs with Bayesian networks to iteratively refine alarm probability models using alarm paths and key statements.

2. Adopt **backward program slicing to extract alarm paths**, preserving critical control/data dependencies and reducing redundant code information.

3. Use task decomposition **to address LLM limitations** (hallucination, context window), enhancing rule learning effectiveness and achieving significant performance gains.

Framework



Framework of BinLLM

Alarm path extractor

- User feedback-based alarm probability models are **limited by manually predefined Datalog rules**, with insufficient expressive power to capture syntactic/semantic features of complex alarms, resulting in poor ability to distinguish related alarms.
- We propose **a syntax-semantics dual-driven fine-grained feature extraction method**, capturing key features from line-level static analysis to support dynamic rule optimization and precise input for LLM-based vulnerability identification.


```
1:  static char *suffixes[] = { "gz", "bz2", "xz", "zip", NULL};
2:
3:  void strip_compression_suffix(const char *name)
4:  {
5:   char *suf = strrchr(name, '.');
6:   if (suf)
7:   {
8:    int suffix_len = strlen(++suf), name_len = 0;
9:    for (int i = 0; compression_suffixes[i]; i++)
10:   {
11:    if (strlen(suffixes[i]) == suffix_len)
12:    {
13:     name_len = strlen(name) - suffix_len -1;
14:     // ...
15:     break;
16:    }
17:   }
18:   if (name_len > 4 &&
          !strncmp(name + name_len - 4, ".tar", 4))
19:    name_len -= 4;
20:   // ...
21:   }
22:   // ....
23: }
```

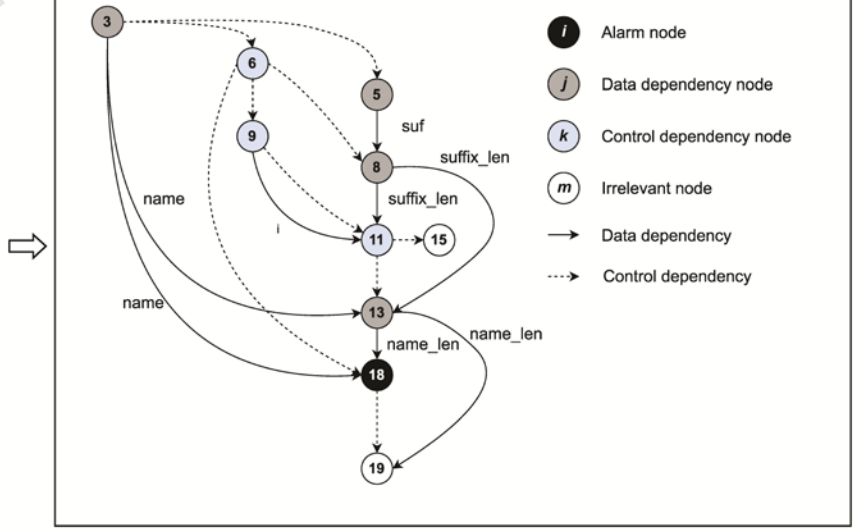
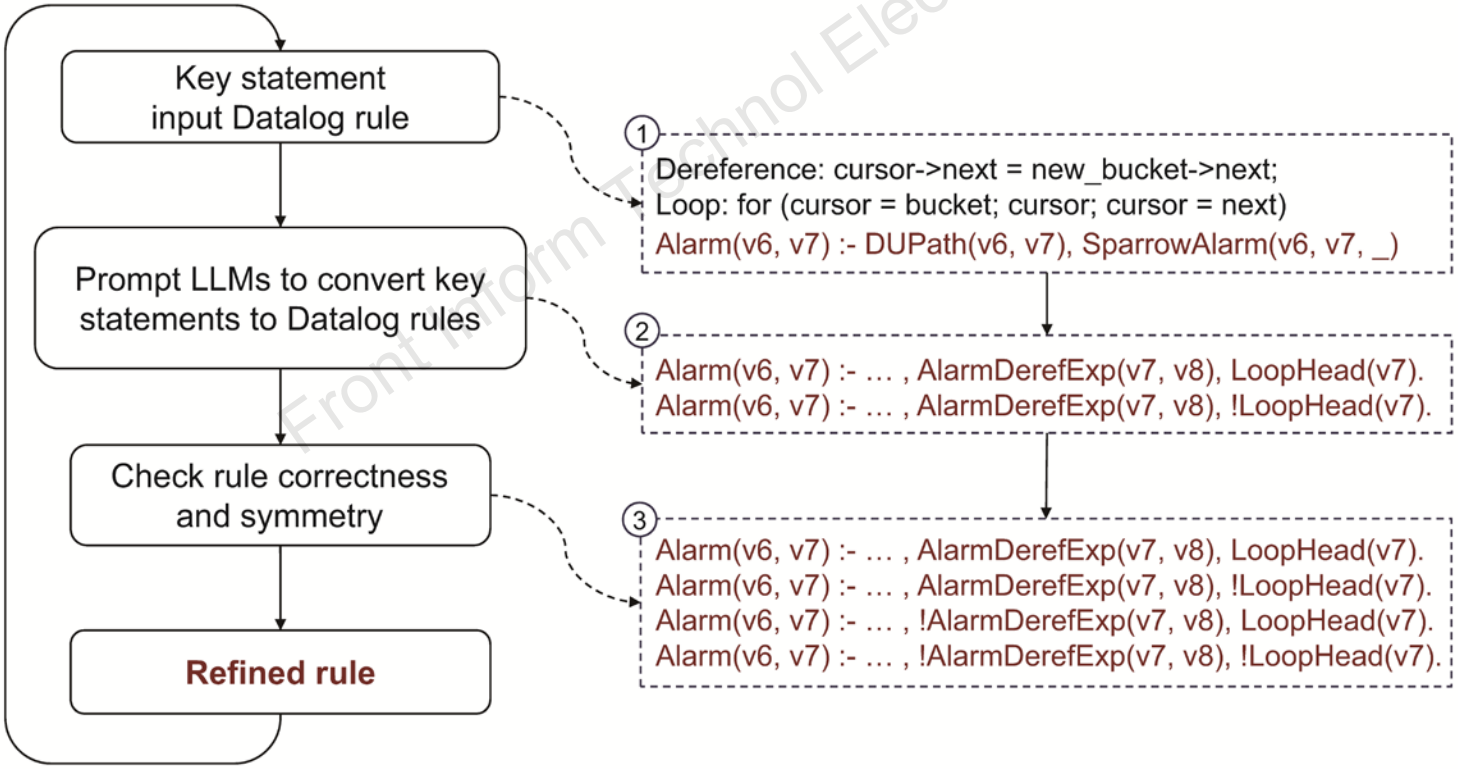


Fig. 2 Example of an intra-procedural backward slice

LLM rule refinement

Due to LLM randomness and insufficient familiarity with Datalog rules, generated rules may violate Datalog inference requirements. We propose **an extended symmetric alarm rule generation method**.



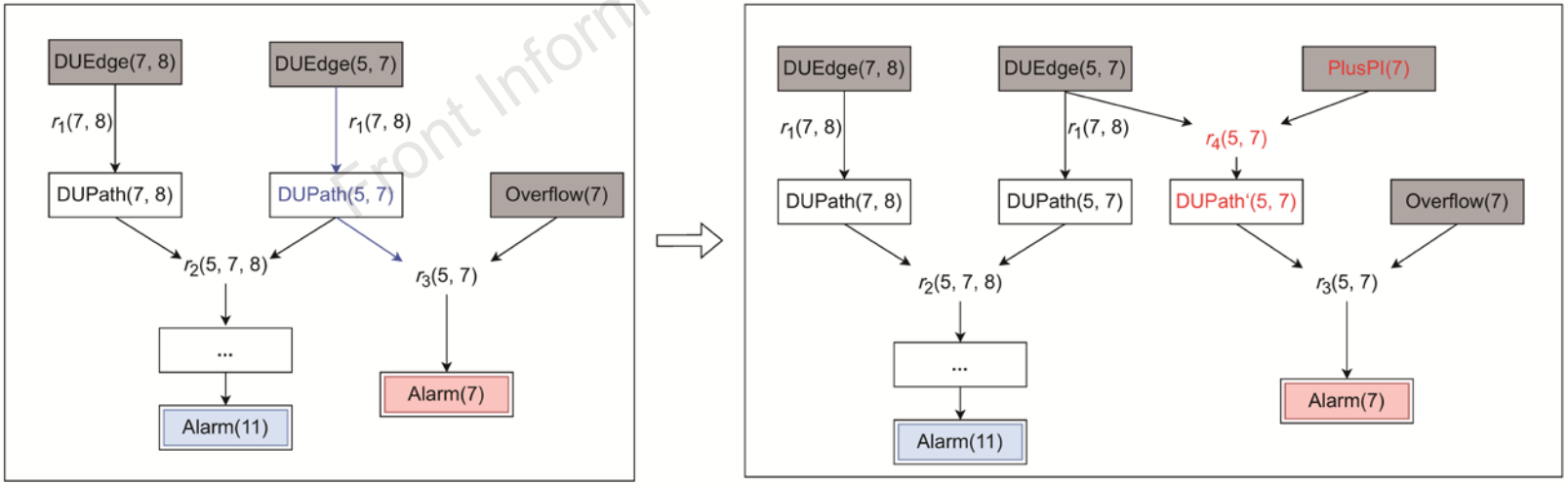
Model refinement

```
1  char *strip_compression_suffix(const
       char *name)
2  {
3   char *s = NULL;
4   size_t len;
5   if(find_compression_suffix(name,&len))
6   {
7    if(strncmp(name+len-4,".tar",4)==0)
8     len -= 4;
9    // ...
10   s = xmalloc (len + 1);
11   memcpy (s, name, len);
12   // ...
13  }
14  return s;
15 }
```

Alarm(7)

LLM rule learning

DUPath(v6, v7) :- DUEdge(v6, v7), PlusPI(v7)
DUPath(v6, v7) :- DUEdge(v6, v7), !PlusPI(v7)



Major results

Table 4 Effectiveness comparison

Program	#Alarms	Bingo	BayeSmith	BinLLM_{N}	BinLLM
cflow-1.5	805	94	60	62	49
fribidi-1.0.7	213	6	3	5	3
grep-2.19	912	53	53	63	37
gzip-1.2.4a	358	145	107	170	125
patch-2.7.1	502	36	34	35	27
sort-7.2	715	176	94	176	15
tar-1.28	1369	218	146	88	86
wget-1.12	891	193	90	193	191
jhead-3.0.0	19	7	4	5	3
optipng-0.5.3	67	14	12	12	9
autotrace-0.31.1	77	77	43	49	34
urjtag-0.8	35	22	17	22	22
ap2s-4.14	27	15	11	7	7
sdop-0.61	150	85	81	77	76
Total	6140	1141	755	964	684

BinLLM_{N} indicates BinLLM without the alarm path extractor. #Alarms reports the number of alarms. The last four columns report the number of iterations until discovering all bugs

Major results

Table 5 Reduction of false generalizations

Program	Bingo			BinLLM		
	Freq	Mag	Freq×Mag	Freq	Mag	Freq×Mag
cflow-1.5	7	22.1	155.0	4	1.0	4.0
fribidi-1.0.7	2	2.0	4.0	1	2.0	2.0
grep-2.19	0	0.0	0.0	0	0.0	0.0
gzip-1.2.4a	191	15.3	2931.8	80	8.9	714.4
patch-2.7.1	4	26.0	104.0	0	0.0	0.0
sort-7.2	7	73.0	511.0	0	0.0	0.0
tar-1.28	27	19.4	523.0	0	0.0	0.0
wget-1.12	122	88.8	10 839.7	94	88.9	8360.4
jhead-3.0.0	0	0.0	0.0	0	0.0	0.0
optipng-0.5.3	3	21.7	65.0	2	39.5	79.0
autotrace-0.31.1	194	12.1	2343.5	152	17.1	2597.7
urjtag-0.8	6	11.0	66.0	14	7.6	106.0
ap2s-4.14	8	4.1	33	1	1	1
sdop-0.61	1552	5.5	8598.1	1396	10.3	14 323
Average	151.6	21.5	1869.6	124.6	12.6	1870.5

Freq and Mag report the number of false generalizations and their average magnitude for the Bayesian program analysis, respectively

Conclusions

1. We propose BinLLM, an innovative Bayesian framework fusing LLMs with static analysis, abstracting Datalog rules via alarm path and key statement extraction.

2. We synergize the strengths of both approaches, leading to enhanced inference quality in limited iterations and reduction in false generalization in alarm ranking.

3. Experimental results demonstrate the effectiveness of BinLLM, achieving 40.1% and 9.4% reduction in the number of checks required for alarm verification compared to two state-of-the-art baselines, Bingo and BayeSmith, respectively, in identifying all vulnerabilities within static analysis alarms.