

논문 2026-3-9 <http://dx.doi.org/10.29056/jsf.2026.09.09>

MSA 서비스 구조 정정을 통한 오픈소스 라이선스 양립성 검증 시스템

박경엽*, 김동완*, 조용준*, 신동명*†

Open Source License Compatibility Verification System Using MSA Service Structure Correction

Kyung-Yeob Park*, Dong-Wan Kim*, YongJoon Joe*, Dong-Myung Shin*†

요 약

클라우드 네이티브 환경에서 마이크로서비스 아키텍처 도입이 가속화됨에 따라 복잡하게 얽힌 오픈소스 라이선스의 양립성을 정확히 검증하는 것의 중요성이 커지고 있다. 그러나 운영 환경에서 확보할 수 있는 설정 정보, 분산 추적 정보, SBOM은 각각 누락과 오류를 포함한다. 본 논문은 설정 정보, 추적 정보, SBOM 등 운영 환경 관측 데이터의 불완전성에 따른 영향을 완화하기 위해 확률적 추론 기반의 서비스 구조 정정 및 충돌 해소 추천 기법을 제안한다. 먼저 이종의 관측 데이터로부터 복수의 구조 후보를 생성하고 관측 적합도와 판정 일관성, 구조 복잡도를 결합한 점수 함수를 통해 최적의 아키텍처를 추천하는 모델을 설계하였다. 이를 바탕으로 정답 구조가 확보된 시나리오 기반 실험을 수행함으로써 제안 기법이 기존 다수결 융합 방식 대비 연결 방식의 정확도를 향상시키고 구조 편집 거리를 감소시켜 실효성 있는 라이선스 검증 시스템으로서 적용 가능성을 확인한다.

Abstract

With the accelerating adoption of microservice architectures in cloud-native environments, accurately verifying the compatibility of complex open-source licenses has become increasingly critical. However, configuration data, distributed traces, and SBOMs available in production environments each contain omissions and errors. To mitigate the effects of incomplete observational data, this paper proposes a probabilistic inference-based service structure correction and conflict resolution recommendation technique. We first generate multiple structure candidates from heterogeneous observations and design a model that selects the highest-scoring structure using a scoring function combining observation fitness, verdict consistency, and structural complexity. Through scenario-based experiments with known ground truths, we confirm that the proposed technique improves connection-type accuracy and reduces graph edit distance compared to the conventional majority voting approach, demonstrating its applicability to practical license compatibility verification.

한글키워드 : 마이크로서비스 아키텍처, 오픈소스 라이선스, 양립성, SBOM, 의존성 추론

keywords : Microservice Architecture, Open Source License, Compatibility, SBOM, Dependency Inference

* 엘에스웨어(주) 소프트웨어연구소 연구개발본부

† 교신저자: 신동명(email: roland@lsware.com)

접수일자: 2026.09.04. 심사완료: 2026.09.10.

게재확정: 2026.09.20.

1. 서론

마이크로서비스 아키텍처 (Microservice Architecture, MSA)는 하나의 애플리케이션을 독립적으로 배포 가능한 다수의 서비스로 분할하여 개발 및 운영하는 방식으로 클라우드 네이티브 환경에서 널리 채택되고 있다[1]. 각 서비스는 개발 생산성을 높이기 위해 다수의 오픈소스 구성요소로 이루어지며 이에 따라 하나의 시스템 안에 여러 종류의 오픈소스 라이선스가 공존하게 된다.

오픈소스 라이선스가 명시하는 의무의 전과 범위는 구성요소가 결합되는 방식에 따라 달라진다. 예를 들어 동일한 실행 단위 안에서 링크된 구성요소는 결합 저작물의 일부로 취급되어 카피레프트(Copyleft) 의무가 전체에 미칠 수 있는 반면 프로세스가 분리된 REST 호출이나 메시지 큐를 경유하는 결합은 별개의 프로그램으로 보아 전과 범위가 제한되는 것으로 해석된다[2][3]. 또한 네트워크를 통한 서비스 제공 행위를 의무 발생 요건으로 규정하는 라이선스도 존재한다. 따라서 MSA 환경에서의 라이선스 양립성을 판정하기 위해서는 각 서비스와 구성요소 간의 결합 방식 및 전체적인 서비스 구조를 파악하는 작업이 선행되어야 한다.

그러나 운영 환경에서 서비스 구조를 파악하기 위해 활용되는 관측 데이터는 불완전성이 존재한다. 배포 설정 정보는 선언된 내용과 실제 런타임 동작이 일치하지 않을 수 있고 분산 추적은 표본 추출(Sampling)로 인해 일부 호출 관계가 누락될 수 있으며 소프트웨어 자재 명세서 (Software Bill of Materials, SBOM)는 생성 도구와 시점에 따라 식별되는 구성요소 및 라이선스가 달라질 수 있다.

이에 본 논문에서는 불완전한 배포 설정 정보, 분산 추적 및 SBOM 데이터를 바탕으로 복수의

서비스 구조 후보를 생성하고 이 중 라이선스 양립성 판정 결과가 가장 일관된 구조를 선택하여 재판정을 수행하는 구조 정정 기법과 검출된 충돌에 대한 해소 전략 추천 기법을 제안한다. 제안 기법의 유효성을 검증하기 위해 불완전한 관측 데이터로부터 서비스 관계와 연결 방식을 정확히 정정할 수 있는지, 정정된 구조의 라이선스 양립성 판정 성능과 관측 결측률 변화에 따른 성능 특성은 어떠한지, 그리고 추천된 충돌 해소 전략이 실제 충돌 서비스 수를 감소시키는지를 평가한다. 본 논문의 구성은 다음과 같다. 2장에서는 오픈소스 라이선스 양립성과 제안 기법에 활용되는 요소 기술을 서술하고 3장에서는 제안 기법의 전체 구조와 세부 알고리즘을 기술한다. 4장에서는 실험을 통해 제안 기법의 유효성을 평가하며 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구

2.1 오픈소스 라이선스 양립성

오픈소스 라이선스 양립성은 서로 다른 라이선스가 적용된 구성요소를 하나의 결과물로 결합할 때, 각 라이선스가 규정하는 의무를 동시에 이행할 수 있는지를 판단하는 문제이다. 두 라이선스가 서로 충돌하는 의무를 규정할 경우, 결합된 결과물은 어느 한쪽의 조건을 위반하게 되므로 적법한 배포가 불가능해진다. 예를 들어 GPL-2.0은 결합 저작물 전체에 동일한 조건의 배포를 요구하면서 추가적인 제한을 부과하는 것을 허용하지 않는 반면 Apache-2.0은 특허 소송을 제기하면 특허 실시권이 종료되는 조항을 포함한다. 이 조항이 GPL-2.0이 금지하는 추가 제한으로 해석되므로 두 라이선스의 구성요소를 하나의 실행 단위로 링크하여 배포하면 어느 한쪽의 조건을 충족할 수 없게 된다[3]. 양립성 판단의 핵심 기준은 결합된 구성요소들이 하나의 저

작물로 취급되는지 여부이며 이는 구성요소 간의 결합 방식에 따라 결정된다.

오픈소스 라이선스는 의무의 강도에 따라 크게 허용형, 강한 카피레프트, 약한 카피레프트의 세 가지로 분류된다. 이러한 라이선스 유형은 소스코드 공개, 재라이선싱, 결합 방식에 따른 의무 전과 범위 등에서 차이가 있으며, 양립성 판단 시에는 이러한 법적·기술적 조건과 소프트웨어의 결합 및 배포 문맥을 함께 고려할 필요가 있다[3]. 추가적으로 네트워크 조항이 포함된 라이선스의 경우 소프트웨어를 직접 배포하지 않고 네트워크 서비스로 제공하는 행위만으로도 소스코드 공개 의무가 발생하도록 규정함으로써 전통적인 배포 개념이 적용되지 않는 서비스형 소프트웨어(Software as a Service, SaaS) 환경까지 의무 범위를 확장하고 있다[4]. 이처럼 동일한 구성요소라도 시스템 내에서 어떤 방식으로 결합되고 노출되는지에 따라 라이선스 의무 범위가 달라진다. 따라서 정확한 양립성 판정을 위해서는 단순한 구성요소 목록을 넘어 전체적인 서비스 구조와 결합 정보가 필수적이다.

하지만 기존의 양립성 판정 자동화 도구들은 이러한 결합 구조를 동적으로 파악하기보다는 주로 소스 코드나 패키지 메타데이터에서 라이선스를 식별한 뒤 사전에 정의된 양립성 표(Matrix)에 대조하는 정적인 방식으로 동작한다. 도구마다 인식할 수 있는 라이선스의 범위와 판정 기준이 상이하여 동일 대상에 대해서도 결과가 달라질 수 있으며 배포된 산출물 검증 시 바이너리 분석을 통해 미신고 구성요소를 식별하는 기법이 활용되기도 하지만 이 역시 정적 분석 기법 중 하나이다[5][6]. 또한 이러한 도구들은 분석 대상의 구성요소 목록과 결합 구조가 사전에 온전히 주어져 있다는 전제하에 동작하므로 서비스가 독립적으로 배포되고 런타임에 결합되는 MSA 환경에서는 해당 전제가 성립하기 어렵다는 근본적

인 한계를 지닌다.

2.2 SBOM

SBOM은 하나의 소프트웨어를 구성하는 모든 구성요소와 그 의존 관계를 기계 판독이 가능한 형식으로 기술한 목록이다. 이는 포함된 구성요소의 구체적인 버전과 적용된 라이선스 정보를 단일 문서에서 통합적으로 파악할 수 있도록 지원한다. SBOM을 구성하는 최소한의 요소로는 공급자, 구성요소명, 버전, 고유 식별자, 의존 관계, 작성자, 생성 시각이 포함되어 있다[7]. SBOM의 대표적 표준 형식으로는 SPDX와 CycloneDX가 있다. SPDX는 구성요소를 패키지 와 파일 단위로 기술하고 구성요소 사이의 관계를 관계 유형으로 표현한다. 이러한 관계 유형에는 포함이나 의존과 같은 일반적인 항목과 함께 정적 링크 및 동적 링크와 같은 결합 방식이 포함된다. 또한 라이선스를 임의의 문자열이 아닌 표준 식별자로 표기하고 복수의 라이선스를 논리식으로 결합할 수 있어 라이선스 정보를 기계적으로 처리하기에 적합하다[8]. CycloneDX는 구성요소와 의존성뿐 아니라 외부 서비스와 참조 정보를 함께 기술할 수 있는 형식을 제공하여 서비스 단위로 구성된 시스템의 구성 정보를 표현하는 데 활용된다[9].

SBOM은 라이선스 식별자와 결합 방식 정보를 제공하여 양립성 판정의 기본 입력이 된다. 그러나 빌드 시점에 생성되어 런타임 서비스 호출 관계를 반영하지 못하며, 실제 배포되지 않는 시험/개발용 구성요소가 포함되거나 결합 방식이 모호하게 표기되는 한계가 있다. 이러한 특성으로 인해 SBOM만으로는 MSA 전체의 결합 구조를 확정하기 어렵다[10]. 클라우드 및 MSA 환경의 라이선스 양립성 판단을 위해서는 소프트웨어 종속성뿐 아니라 배포·운용 방식과 서비스 간 통신 관계 등의 추가 정보가 필요하다[11].

2.3 분산 추적

분산 추적은 단일 요청이 다수의 서비스를 거쳐 처리되는 과정을 추적함으로써 서비스 간 호출 관계와 실행 경로를 기록하는 기술이다[12]. 시스템 진입 시점에 전역 식별자를 발급하고 개별 서비스의 처리 구간을 시작과 종료 시각을 갖는 스패ن(Span) 단위로 기록한 후 동일 식별자를 공유하는 스패들을 병합하여 전체 호출 트리로 복원한다. 이는 노드 수가 많고 호출 경로가 복잡한 분산 환경에서도 개별 요청의 이동 경로를 사후에 재구성할 수 있도록 지원한다. 스패의 구조, 서비스 간 전달 규약, 그리고 속성 표기법은 표준 규격으로 정의되며 개별 스패 속성에는 통신 프로토콜 및 대상 서비스 명칭이 포함된다[12]. 특히 분산 추적에서 관측된 호출은 실제 사용 관계에 대한 직접적인 근거를 제공하며 호출에 사용된 통신 방식을 함께 확인할 수 있다는 장점이 있다. 다만 관측되지 않았다는 사실만으로 해당 관계가 사용되지 않는다고 단정할 수는 없다.

반면 모든 요청을 기록할 경우 저장 및 전송 비용의 과도한 오버헤드가 발생하므로 일부 요청만을 선택적으로 기록하는 표본 추출이 주로 사용된다. 이로 인해 호출 빈도가 낮은 경로는 기록에서 누락될 수 있으며 계측(Instrumentation)이 적용되지 않은 서비스를 경유하는 호출 역시 관측되지 않는다. 이러한 문제를 해결하기 위해 시스템 내부 수정 없이 송수신 메시지의 시간 관계만으로 호출 경로를 추론하는 기법도 제시되었으나 이 역시 추론 결과에 확률적 오차가 발생한다는 한계를 지닌다[13]. 결과적으로 분산 추적으로 도출된 구조는 실제 호출 관계를 반영하지만 완전성을 보장하기 어려우며 개별 구성요소 간의 세부 결합 방식까지는 파악할 수 없다는 한계가 있다.

2.4 다중 소스 관측 통합 기법

설정 정보, 분산 추적, 그리고 SBOM은 서로 다른 시점과 관점에서 동일한 시스템을 관측하기 때문에, 동일한 결합 관계에 대해 상이한 결과를 제시하는 경우가 발생한다. 이처럼 복수의 데이터 소스가 상충하는 정보를 제공할 때 참값을 추정하는 문제를 진실 발견(Truth Discovery)이라고 정의한다. 진실 발견의 가장 단순한 접근법은 다수의 소스가 지지하는 값을 채택하는 다수결 방식이며 각 소스의 과거 정확도를 반영하여 신뢰도를 산출하고 이를 가중치로 적용해 최종 값을 결정하는 기법도 활용된다[14].

이러한 기법들은 개별 데이터 소스들이 서로 독립적으로 관측을 수행한다는 가정에 기반한다.

그러나 두 소스가 동일한 원본에서 값을 가져오는 경우에는 오류까지 함께 공유하게 되므로 단순히 다수가 같은 값을 제시한다는 사실이 참값의 근거가 되지 못하며 오히려 오류를 증폭시킬 위험이 있다[15]. 실제 MSA 환경에서는 설정 정보와 SBOM이 동일한 의존성 선언 파일에 기반하여 생성되는 경우가 존재하므로 이러한 데이터 소스 간의 종속성 문제가 빈번하게 발생한다. 또한 기존 기법들은 개별 관측값의 일치도만을 기준으로 채택 여부를 결정한다. 서비스 구조처럼 개별 데이터가 독립적으로 존재하기보다 전체 그래프의 형태로 후속 작업에 활용되는 경우 특정 값의 선택이 후속 작업의 최종 결과에 미치는 영향은 고려되지 않는다.

3. 서비스 구조 정정 및 충돌 해소 추천 기법

3.1 시스템 개요

제안 시스템은 그림 1과 같이 네 개의 계층으로 구성된다. 첫 번째로 관측 데이터 수집 및 정규화 계층은 배포 설정 정보, 분산 추적, SBOM 데이터를 수집하여 통일된 관측 형식으로 변환한

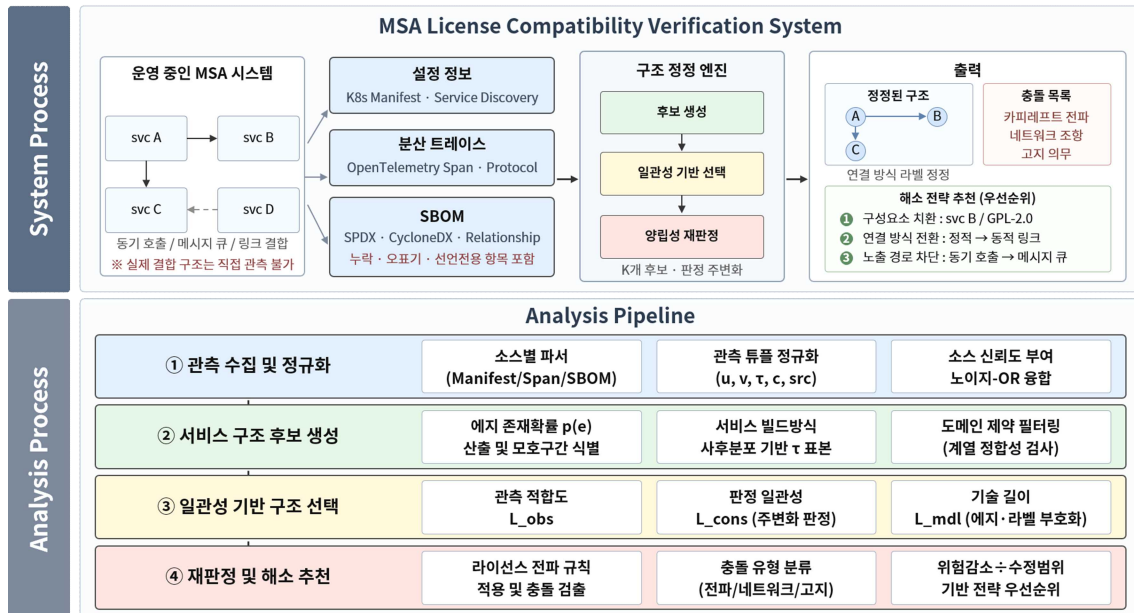


그림 1 제안 시스템 개요도

Fig. 1. Overview of the Proposed System

다. 두 번째로 서비스 구조 후보 생성 계층은 정규화된 관측 데이터로부터 복수의 서비스 구조 후보군을 생성하며 세 번째, 일관성 기반 구조 선택 계층은 생성된 후보들을 평가하여 최적의 단일 구조를 확정한다. 마지막으로 양립성 재판정 및 충돌 해소 추천 계층은 선택된 구조에 라이선스 전파 규칙을 적용하여 양립성 충돌 여부를 재판정하고 이에 대한 구체적인 해소 전략을 추천한다.

제안 시스템에서 정정 대상이 되는 서비스 구조는 방향 그래프 $G=(V, E, \tau, l, V_G)$ 로 정의한다. V 는 서비스와 오픈소스 구성요소의 집합, E 는 의존 관계의 집합이며 τ 는 각 예지의 연결 방식 (e.g., 정적 링크, 동적 링크, 프로세스 분리 호출 등)을 나타내는 함수이다. l 은 각 구성요소에 SPDX 라이선스 식별자를 대응시키는 함수이며 V_G 는 외부 요청이 진입하는 게이트웨이 서비스의 집합이다. 연결 방식은 카피레프트 전파 범위를 결정하는 주요 판정 요소이며 네트워크 조항

의 적용 여부는 V_G 로부터의 도달 가능성과 함께 판정한다. 본 연구의 단순화된 판정 규칙에서는 외부 진입 서비스로부터 동기 호출만으로 이어지는 경로가 존재하는 서비스를 외부 노출로 정의하며, 메시지 큐와 같은 비동기 서비스 경계는 해당 노출 경로에서 제외한다. 예지의 존재 여부와 연결 방식이 정정 대상이 된다. 또한 정답 구조는 G_T , 관측을 모두 합집합한 구조는 G_O , 제안 기법이 선택한 구조는 G_S 로 표기한다. 또한 V 의 부분 집합 중 서비스만으로 이루어진 집합을 V_S , 구성요소만으로 이루어진 집합을 V_C 로 표기한다. l 과 V_G 는 후보에 관계없이 고정된 입력이므로 k 번째 후보 구조는 $G_k=(V, E_k, \tau_k, l, V_G)$ 로 표기한다. 예지는 서비스 간 호출과 서비스-구성요소 결합의 두 유형으로 구성되며 구성요소 사이의 관계는 본 연구에서 모델링하지 않는다. 본 연구는 서비스 및 구성요소 식별자 집합이 사전에 확보된 환경을 전제로 하며 관측 간 불일치가 발생하는 의존 관계의 존재 여부와 연결 방식의 정정

을 대상으로 한다.

3.2 관측 데이터 수집 및 정규화

관측 데이터 수집 및 정규화 계층에서는 세 종류의 데이터 소스로부터 정보를 수집한다. 구체적으로, 배포 설정 정보에서는 Kubernetes 매니페스트와 서비스 디스커버리 등록 정보로부터 선언된 서비스 간 관계를 추출하고 빌드 설정과 의존성 선언 파일로부터 서비스와 구성요소 간 결합 관계를 추출하며 분산 추적 데이터에서는 OpenTelemetry 스패ن으로부터 실제 호출 관계와 프로토콜 정보를 추출한다. 또한 SBOM에서는 서비스별 구성요소와 라이선스 식별자, 관계 유형을 추출한다. 수집된 모든 관측 데이터는 (출발 정점, 도착 정점, 연결 방식, 신뢰도, 출처) 형태의 관측 튜플로 정규화된다. 이때, 동일한 관계에 대해 상충하는 관측값이 존재하더라도 이를 하나로 병합하지 않고 그대로 유지하여 다음 단계인 후보 생성에 활용한다. 표 1은 이러한 각 관측 소스의 특성을 나타낸다.

표 1. 관측 소스별 특성

Table 1. Characteristics of Observation Sources

구분	설정 정보	추적 정보	SBOM
획득 시점	배포 정의	런타임	빌드
강점	전역 관계 파악	실제 호출 반영	구성요소 라이선스
주요 오류	선언/실제 불일치	표본 추출에 따른 누락	도구별 편차
연결 방식 정보	부분적	프로토콜 수준	관계 유형 수준

3.3 서비스 구조 후보 생성

서비스 구조 후보 생성 계층에서는 모든 관측 튜플을 합집합한 통합 관측 그래프 G_U 를 구성하고 각 에지 e 에 대하여 관측 간 조건부 독립을 근사적으로 가정하는 Noisy-OR 방식을 적용해

관측 신뢰도와 데이터 소스 신뢰도를 결합한 최종 존재 확률 $p(e)$ 를 산출한다. 이는 식(1)과 같이 표현된다.

$$p(e) = 1 - \prod_{r \in R(e)} (1 - c_r \cdot \rho(s_r)) \quad (1)$$

위 식에서 $R(e)$ 는 에지 e 에 대한 관측의 집합, c_r 은 개별 관측의 신뢰도, $\rho(s_r)$ 은 해당 관측을 제공한 데이터 소스의 신뢰도를 의미한다. 산출된 존재 확률 $p(e)$ 가 상한 이상인 에지는 항상 구조에 포함하고 하한 미만인 에지는 항상 제외하며 그 사이의 모호 구간에 속하는 에지에 대해서만 확률적 표본 추출을 수행한다. 최종적인 후보 집합은 서로 다른 임계값 기준을 적용하여 얻은 3개의 결정적 구조와 K-3개의 확률적 표본으로 구성된다. 다만 데이터 소스 간 종속성은 이러한 독립 근사만으로 제거되지 않으므로 제안 기법은 소스별 신뢰도 가중과 3.4절의 판정 일관성 평가를 통해 그 영향을 완화한다. 또한 후보의 에지 집합은 통합 관측 그래프의 에지 집합으로 제한되어 모든 후보에 대해 $E_k \subseteq E_U$ 가 성립하므로 세 관측 소스에서 모두 누락된 참 에지는 본 기법의 정정 대상에 포함되지 않는다. 본 실험 데이터에서 이러한 에지는 전체 참 에지의 8.5%이며 이는 관측된 모든 에지를 수용하는 Union graph의 재현율 상한을 0.915로 제한한다. 신규 에지의 탐색은 별도의 링크 예측 문제로 본 연구의 범위에서 제외한다.

에지의 존재 여부뿐만 아니라 구체적인 연결 방식 또한 관측 데이터만으로는 명확히 식별되지 않는 경우가 존재한다. 실제로 SBOM은 관계 유형을 구체화하지 않고 포괄적인 의존 관계로만 표기하는 경우가 있고 배포 설정 정보 또한 정확한 결합 방식을 특정하기 어렵다. 제안 시스템에선 단일 서비스 내에 포함된 구성요소들이 유사

한 빌드 환경을 공유한다는 특성을 활용하여 이러한 문제점을 보완한다. 제안 기법은 관측된 레이블로부터 서비스별 정적·동적 링크 비율의 사후 분포를 추정하고 이를 사전 확률로 활용하여 결합 방식이 명시되지 않은 에지의 연결 방식을 표본 추출한다. 이후 추출된 연결 방식에 도메인 제약 규칙을 적용하여 서비스와 구성요소 간 결합은 링크 계열로 서비스 간 호출은 프로세스 분리 계열로 매핑한다. 전체 후보 생성 알고리즘은 표 2와 같다.

표 2. 구조 후보 생성 알고리즘

Table 2. Structure Candidate Generation Algorithm

Algorithm 1: Candidate Structure Generation

Require: Observations $O = \{u, v, \tau, c, src\}$,
Vertex set V ,
thresholds $0 \leq lo < 0.5 < hi \leq 1$,
Candidate count $K > 3$

Ensure: Candidate set $C = \{G_1, \dots, G_K\}$

- 1: **Probability Calculation:**
- 2: **for** each edge e in the union graph **do**
- 3: $p(e) \leftarrow 1 - \prod_r (1 - c_r \cdot \rho(s_r))$
- 4: **end for**
- 5: **Prior Probability Estimation:**
- 6: **for** each service v in V_S **do**
- 7: $\pi_v \leftarrow$ static/dynamic link ratio from labels of edges with $p(e) \geq 0.5$
- 8: **end for**
- 9: **Deterministic Construction:**
- 10: **for** $k=1$ to 3 **do**
- 11: $\theta_k \leftarrow$ k-th value of $(lo, 0.5, hi)$
- 12: $E_k \leftarrow \{e : p(e) \geq \theta_k\}$
- 13: $\tau_{k(e)} \leftarrow \text{ArgMaxLink}(\text{evidence}(e), \pi)$
- 14: $G_k \leftarrow (V, E_k, \text{Constrain}(\tau_k))$
- 15: **end for**
- 16: **Stochastic Sampling:**
- 17: **for** $k=4$ to K **do**
- 18: $E_k = \{e : p(e) \geq hi\} \cup \{e : lo \leq p(e) < hi, \text{Bern}(p(e)) = 1\}$
- 19: $\tau_{k(e)} \leftarrow \text{SampleLink}(\text{evidence}(e), \pi)$
- 20: $G_k \leftarrow (V, E_k, \text{Constrain}(\tau_k))$
- 21: **end for**
- 22: **Return Results:**
- 23: **return** results: $C = \{G_1, \dots, G_K\}$

3.4 일관성 기반 구조 선택 및 재판정

일관성 기반 구조 선택 계층에서는 3.3절에서 생성된 각 후보 구조 G_k 에 대하여 식 (2)의 점수 함수를 계산하고 산출된 점수가 최대인 후보를 최적의 구조로 확정한다. 이때, 점수 함수를 구성하는 세 가지 항은 산출 기준이 다르므로 후보 집합 내에서 최소-최대 정규화(Min-Max normalization)를 적용하여 동일한 척도로 변환한 뒤 결합한다.

$$S(G_k) = \alpha \cdot L_{obs} + \beta \cdot L_{cons} + \gamma \cdot L_{mdl} \quad (2)$$

식 (2)의 첫 번째 항인 관측 적합도 L_{obs} 는 후보 구조가 관측과 부합하는 정도를 식 (3)의 가중 로그 적합도로 측정한다. 우변의 첫 두 항은 에지의 존재 여부에 대한 적합도이며 합산 범위는 통합 관측 그래프의 에지 집합 E_U 로 후보에 포함되지 않은 에지는 $E_U \setminus E_G$ 에서 평가한다. 세 번째 항은 연결 방식 레이블의 일치 여부를 관측 신뢰도 c_r 과 소스 신뢰도 $\rho(s_r)$ 로 가중한 것으로 후보의 연결 방식이 관측 레이블과 일치하면 $q=0.9$, 일치하지 않으면 $q=0.1$ 을 부여하며 연결 방식이 기술되지 않은 관측은 이 항에서 제외한다. 로그 계산의 수치 안정성을 위해 $p(e)$ 는 $[\epsilon, 1-\epsilon]$ 구간으로 절단하며 $\epsilon=10^{-3}$ 을 사용한다. 여기서 $R_r(E_G)$ 는 후보 구조 G 에 포함된 에지에 대한 관측 중 연결 방식 레이블이 존재하는 관측의 집합을 의미한다.

$$L_{obs}(G) = \sum_{e \in E_G} \log p(e) + \sum_{e \in E_U \setminus E_G} \log(1-p(e)) \quad (3) \\ + \sum_{r \in R_r(E_G)} c_r \rho(s_r) \log q(\tau_G(e_r), \tau_r)$$

기술 길이 L_{mdl} 은 불필요한 복잡성을 배제하고 간결한 모델을 선호하는 최소 기술 길이 원리

를 참고한 구조 복잡도 항 식 (4)와 같이 정의한다. 부호화 비용은 예지 1개당 1비트와 서비스별 연결 방식 레이블의 엔트로피 부호화 비용의 합으로 산출하며 단일 서비스 내부의 결합 방식이 이질적일수록 더 큰 페널티를 부여한다. 이때 $n_v(G)$ 는 후보 G 에서 서비스 v 에 결합된 구성요소의 수, $\hat{\pi}_v(G)$ 는 후보 G 에서 산출한 해당 서비스의 정적·동적 링크 비율이다. 3.3절의 사전 분포는 관측으로부터 추정된 고정값인 반면 이 항은 후보마다 다시 계산한 분포를 사용하며 $n_v(G)=0$ 인 서비스는 엔트로피 항을 0으로 둔다. 합산 범위는 서비스 집합 V_S 이며 구성요소 사이에는 연결 방식 레이블이 정의되지 않으므로 부호화 비용 산정 대상에서 제외한다.

$$L_{mdl}(G) = -(|E_G| + \sum_{v \in V_S} n_v(G) \cdot H(\hat{\pi}_v(G))) \quad (4)$$

판정 일관성 L_{cons} 는 후보 집합을 구조 사후 분포의 표본으로 간주하여 정의한다. 각 후보에 라이선스 전파 규칙 집합 R 을 적용하여 충돌 서비스 집합 $D(G)$ 를 산출하고 충돌로 판정된 비율을 식 (5)와 같이 서비스 s 의 주변 충돌 확률로 추정한다. 이때, 평균의 대상은 전체 후보 집합 C 가 아니라 확률적으로 표본 추출된 부분 집합 $C_s = \{G_1, \dots, G_K\}$ 로 한정한다. 서로 다른 임계값으로부터 도출한 3개의 결정적 구조는 특정 임계 규칙에 기반하므로 이를 평균에 포함하면 주변 확률이 임계값 선택에 영향을 받기 때문이다.

$$P(s) = \frac{|\{G \in C_s : s \in D(G)\}|}{|C_s|}, \quad (5)$$

$$M(s) = I[P(s) \geq t]$$

주변 판정 $M(s)$ 는 $P(s)$ 가 임계값 t 이상인 서비스를 충돌로 분류한 것이며 각 개별 후보의

판정 결과가 이 주변 판정과 일치하는 비율이 식 (6)의 L_{cons} 로 정의된다.

$$L_{cons}(G_k) = \frac{|\{s \in V_S : I[s \in D(G_k)] = M(s)\}|}{|V_S|} \quad (6)$$

확률이 가장 높은 단일 구조가 내리는 판정 결과와 구조의 불확실성을 주변화한 판정 결과는 상이할 수 있으므로 해당 기법은 점수가 최대인 후보를 최종 구조 G_S 로 선택한다. 임계값 t 는 예비 실험에서 판정 F1을 최대화하는 값으로 탐색하였으며 0.5보다 낮은 값으로 설정한다.

표 3. 구조 선택 및 양립성 재판정 알고리즘

Table 3. Structure Selection and Compatibility Re-verification Algorithm

Algorithm 2: Consistency-based Selection

Require: Observations O , Candidates C , stochastic subset C_s , rules R , weights α, β, γ , thresholds

t

Ensure: Selected structure G_S , verdicts D

1: **Marginal Probability Estimation:**

2: $D_k = D(G_k) \leftarrow \text{ApplyPropagationRules}(G_k, R)$,
all k

3: **for** each service s in V_S **do**

4: $P(s) \leftarrow |\{G_k \in C_s : s \in D_k\}| / |C_s|$

5: $M(s) \leftarrow [P(s) \geq t]$

6: **end for**

7: **Score Calculation & Normalization:**

8: **for** $k=1$ to K **do**

9: $f_k \leftarrow \text{LogLik}(O|G_k)$

10: $c_k \leftarrow \text{Agreement}(D_k, M)$

11: $m_k \leftarrow -(|E_k| + \text{LabelCost}(G_k))$

12: **end for**

13: min-max normalize f, c, m over $k=1..K$

14: $S(G_k) \leftarrow \alpha \cdot f_k + \beta \cdot c_k + \gamma \cdot m_k$, all k

15: **Final Selection:**

16: $G_S \leftarrow \arg\max_k S(G_k)$

17: $D \leftarrow \text{ApplyPropagationRules}(G_S, R)$

18: **Output results:** G_S, D

이는 소수의 후보에서만 나타나는 충돌도 검

토 대상으로 남기는 설정으로 미탐 비용이 오탐의 처리 비용보다 큰 컴플라이언스 환경의 특성과 부합한다. 최종적으로 선택된 구조 G_5 에 규칙 R 을 적용한 결과가 제안 시스템의 최종 판정이 된다. 표 3은 이러한 구조 선택 및 양립성 제판정의 전체 알고리즘을 나타낸다.

3.5 충돌 해소 추천

제판정 결과 검출된 라이선스 충돌은 크게 세 가지 유형으로 분류되며 시스템은 각 유형에 최적화된 해소 전략 후보를 제시한다.

- **카피레프트 전파 충돌 (Copyleft Propagation Conflict):** 카피레프트 구성요소와 비양립 구성요소가 동일한 결합 단위에 포함된 경우 발생한다. 이를 해소하기 위해 결합 방식을 프로세스 분리로 변경하거나 문제가 되는 요소를 양립 가능한 구성요소로 치환하는 전략을 제시한다.
- **네트워크 조항 충돌 (Network Clause Conflict):** AGPL 등 해당 라이선스의 구성요소가 외부로 제공되는 서비스 경로에 포함된 경우로 해당 서비스의 소스코드를 공개하거나 구성요소를 치환하는 전략, 그리고 게이트웨이로 부터의 동기 호출 경로를 메시지 큐 등 비동기 경계로 전환하여 본 연구의 판정 규칙상 외부 노출 경로를 차단하는 전략이 생성된다.
- **고지 의무 미이행 위험 (Notice Obligation Risk):** 라이선스 고지 파일의 보완 및 배포물 구성 상태에 대한 점검 가이드를 제시한다.

각 해소 전략의 추천 우선순위는 해당 전략을 적용했을 때 감소하는 충돌 서비스의 수와 구조 변경에 요구되는 수정 범위의 비율을 기반으로 정량 산정한다. 이때, 수정 범위는 변경 대상이

되는 정점 및 에지의 수로 근사한다. 단, 라이선스 제협상이나 듀얼 라이선싱과 같이 단순 기술적 조치만으로 결정할 수 없는 사항은 자동 추천 대상에서 제외하고 관리자의 인적 검토가 필요한 항목으로 별도 표시한다.

4. 제안 시스템 실험 및 평가

4.1 실험 환경 및 데이터셋

제안 기법의 성능을 정량적으로 평가하기 위해서는 정답 구조가 명확히 확보된 환경이 필요하므로 300개의 MSA 가상 시나리오를 서로 다른 난수 시드로 독립 생성하여 실험 데이터셋을 구축하였다. 데이터셋의 각 시나리오는 다수의 서비스와 오픈소스 구성요소로 이루어지며 실험에서는 12종의 라이선스를 사용하고 허용형 77%, 카피레프트 계열 및 네트워크 조항 23%로 구성하였다.

표 4. 실험 데이터셋 및 관측 설정
Table 4. Experimental Dataset and Observation Settings

항 목	값
시나리오 수	300개 (독립 생성)
서비스/구성요소	서비스 5~30개, 서비스당 구성요소 3~12개
라이선스	SPDX 식별자 기반 12종
빌드 방식 동일성	서비스 내 결합 85% 동일
설정 정보	서비스 관계 80% 관측, 선언 전용 30%
추적 정보	서비스 관계 70% 관측, 구성요소 미관측
SBOM	구성요소 85% 관측, 45% 관계 유형 미기술
연결 방식 오프기	설정 서비스 관계 10%, 설정 구성요소 25%, 추적 정보 5%, SBOM 10%

또한 시스템에서 수집되는 3종의 관측 데이터(설정 정보, 추적 정보, SBOM)는 실제 환경의 불확실성을 모사하기 위해 고의적인 누락, 오류

기, 그리고 소스 간 상관관계에 의한 허위 관측을 포함하도록 구성하였다. 상세한 실험 데이터셋의 규모와 관측 노이즈 설정은 표 4와 같다.

판정에 사용한 라이선스 전파 규칙은 결합 방식(e.g., 정적·동적 링크, 프로세스 분리 등)에 따른 의무 발생 여부로 단순화하여 적용하였다. 단, 3.5절에서 정의한 충돌 유형 중 고지 의무 미이행 위험은 아키텍처 구조의 연결성에 의존하지 않는 독립적인 위반 사항이므로 본 정량 평가 대상에서는 제외하였다.

4.2 비교 기법 및 평가 지표

제안 기법의 성능을 객관적으로 검증하기 위해 다음과 같은 5가지 베이스라인 기법을 비교군으로 설정하였다.

- **Config only / Trace only:** 각각 설정 정보와 추적 정보 단일 소스만을 사용하여 구조를 구성한다.
- **Union graph:** 관측된 모든 에지를 포함하며, 연결 방식은 신뢰도가 가장 높은 관측을 채택한다.
- **Majority voting:** 해당 에지를 관측할 수 있는 소스의 과반수가 보고한 경우에만 에지를 채택한다.
- **Confidence only:** 융합된 에지 존재 확률이 특정 임계값 이상인 경우만 채택한다.

제안 기법 및 비교군의 최적 하이퍼파라미터는 본 실험과 분리된 15개의 예비 시나리오를 통해 판정 F1 점수를 최대화하는 값으로 탐색하였다. 그 결과 Confidence only의 임계값은 0.55로 설정하였고 제안 기법은 후보 수 $K=50$, 가중치 $\alpha=0.5$, $\beta=0.2$, $\gamma=0.3$, 주변 판정 임계값 $t=0.20$ 으로 설정하여 Python 3.11 환경에서 실험을 수행하였다. 평가 지표는 크게 구조 정정과 양립성 판정의 두 축으로 구성된다.

- **구조 정정 성능:** 정답 구조 대비 에지 집

합의 관계 F1 점수, 정답 구조와 일치하는 에지의 연결 방식 정확도(Accuracy), 그리고 Union graph 대비 그래프 편집 거리(Graph Edit Distance, GED) 감소율(GEDR)을 측정한다. 정점 집합은 고정되므로 GED는 에지의 삽입, 삭제, 치환 비용의 합으로 산출하며 감소율은 식 (7)과 같이 정의한다. (음수일 경우 정답에서 더 멀어졌음을 의미한다.)

$$GEDR = \frac{d_u - d_s}{d_u} \times 100, \quad d_x = GED(G_x, G_T) \quad (7)$$

- **양립성 판정 성능:** 서비스별 충돌 존재 여부(양성)를 정답과 비교하여 Accuracy, Precision, Recall, F1 점수로 측정한다. 전체 300개 시나리오의 누적 서비스 중 실제 충돌이 존재하는 양성 비율은 24.3%로 데이터 불균형이 존재한다. 따라서 Accuracy는 이 비율에 크게 의존하므로 기법 간의 실질적 성능 비교에는 F1 점수를 핵심 지표로 함께 참조한다.

4.3 실험 결과

표 5의 구조 정정 성능을 살펴보면 단일 소스 기법과 Majority voting은 관측 한계가 누적되어 전체 그래프 편집 거리 감소율이 음수로 나타나는 등 정답 구조에서 오히려 멀어졌다. 반면 제안 기법은 Union graph 대비 관계 F1 점수는 0.879로 근접하게 유지하면서도 연결 방식 정확도를 0.808에서 0.888로 향상시켜 전체 GED를 13.9% 감소시켰다. 즉, 일부 정답 에지를 누락하는 대신 연결 레이블을 정확하게 정정하여 전체 편집 거리를 최적화했음을 알 수 있다. 한편 Trace only의 연결 방식 정확도가 0.950으로 가장 높은 것은 이 지표가 정답과 일치하는 에지만 적용되기 때문이며

Trace only는 전체 예지의 약 10%만 구조에 포함하므로 관계 F1과 함께 해석해야 한다.

표 5. 구조 정정 성능 비교

Table 5. Comparison of Structure Correction Performance

비교 기법	관계 F1	연결 방식 정확도	GED 감소율 (%)
Config only	0.594	0.788	-84.6
Trace only	0.189	0.950	-130.9
Union graph	0.892	0.808	0.0
Majority voting	0.529	0.856	-82.4
Confidence only	0.870	0.808	-4.9
Proposed	0.879	0.888	13.9

양립성 판정 성능을 나타낸 표 6에서 보수적인 예지 채택 전략을 취한 비교군들은 정밀도는 높으나 재현율이 급감하여 전반적인 F1 점수가 저조했다. 그러나 제안 기법은 관측 누락으로 놓치던 충돌까지 추가로 검출하며 가장 높은 재현율(0.878)과 판정 F1 점수(0.864)를 달성했다. 부트스트랩 5,000회에서 판정 F1 차이는 Union graph 대비 0.010(95% 신뢰구간 0.001~0.019), Confidence only 대비 0.034(0.023~0.046)였으며, 시나리오 단위 대응표본 t-검정($t=2.23$, $df=299$, $p=0.026$)에서도 같은 결론을 얻었다.

표 7은 점수 함수를 구성하는 세 항의 기여도를 분석한 결과이다. 한 항의 가중치를 0으로 두고 나머지 두 항의 합이 1이 되도록 재정규화하

여 측정하였다. 각 제거실험에서는 동일한 관측 데이터와 후보 집합을 사용하고, 제거 대상 항의 가중치와 이에 따른 나머지 가중치의 재정규화만 변경하였다.

표 6. 양립성 판정 성능 비교

Table 6. Comparison of Compatibility Verdict Performance

비교 기법	Acc.	Prec.	Rec.	F1
Config only	0.815	0.859	0.284	0.427
Trace only	0.757	-	0.000	0.000
Union graph	0.929	0.857	0.852	0.854
제안 기법 예지 (레이블 Union)	0.927	0.863	0.833	0.847
Majority voting	0.808	0.900	0.235	0.373
Confidence only	0.921	0.876	0.788	0.830
Proposed	0.933	0.850	0.878	0.864

관측 적합도 항을 제거하면 관계 F1이 0.160으로 낮아지고 GED 감소율이 -133.6%까지 떨어져 이 항이 후보 선택의 기본 신호임을 확인하였다. 판정 일관성 항을 제거하면 구조 정정 지표는 거의 변하지 않으나 판정 F1이 0.864에서 0.851로 낮아져 이 항이 라이선스 판정에 중요한 구조적 차이를 고려한 후보 선택에 기여하는 것으로 해석된다. 반면 구조 복잡도 항은 제거했을 때 판정 F1이 0.870으로 오히려 높아졌다. 이는 해당 항의 가중치가 15개의 예비 시나리오에 과적합되었을 가능성을 시사하며, 본 논문은 평가

표 7. 점수 함수 구성 항 제거에 따른 성능 변화

Table 7. Performance Changes after Removing Each Scoring Term

비교 기법	가중치 (α, β, γ)	관계 F1	연결 방식 정확도	GED 감소율 (%)	판정 F1
Proposed (전체)	(0.50, 0.20, 0.30)	0.879	0.888	13.9	0.864
관측 적합도 항 제거	(0.00, 0.40, 0.60)	0.160	0.916	-133.6	0.142
판정 일관성 항 제거	(0.62, 0.00, 0.38)	0.878	0.888	13.5	0.851
구조 복잡도 항 제거	(0.71, 0.29, 0.00)	0.881	0.888	14.4	0.870

표 8. 연결 방식 유형별 정확도
Table 8. Linkage Type Accuracy by Category

연결 방식	대상 예지수	Union graph	Proposed	개선
정적 링크	14,474	0.632	0.858	+0.226
동적 링크	19,825	0.895	0.891	-0.004
프로세스 분리	4,293	0.945	0.959	+0.014
메시지 큐	1,803	0.933	0.933	+0.000

집합을 이용한 사후 조정을 피하기 위해 예비 실험에서 선택된 가중치를 그대로 유지하였다.

표 8은 연결 방식을 유형별로 나누어 분석한 결과이다. 평가 대상은 제안 기법이 선택한 구조와 정답 구조에 공통으로 포함된 예지 40,395개이며, 연결 방식 유형은 정답 레이블을 기준으로 구분하였다. 제안 기법의 예지 집합은 Union graph의 예지 집합에 포함되므로, 해당 예지는 두 기법과 정답 구조에 모두 포함된다. 이에 두 기법을 동일한 예지 집합에서 평가하고, 유형별 정확도는 전체 시나리오에서 누적한 해당 유형의 레이블 일치 건수를 해당 유형의 평가 예지 수로 나누어 산출하였다. 표 5는 시나리오별 정확도의 평균을, 표 8은 전체 평가 예지를 합산한 비율을 제시한다. 집계 방식은 다르지만 표 8의 전체 정확도는 Union graph 0.808, 제안 기법 0.888로 표 5의 값과 소수점 셋째 자리까지 일치하였다. 연결 방식별로는 정적 링크 정확도가 Union graph

대비 0.226 상승한 반면 동적 링크 정확도는 0.004 낮아져 개선이 정적 링크에 집중되었다. 제안 기법의 연결 방식 오분류를 동일한 평가 대상에서 집계한 결과, 전체 4,519건 가운데 정적 링크와 동적 링크 사이의 혼동은 4,221건이었으며 프로세스 분리와 메시지 큐 사이의 혼동은 298건이었다. 판정 성능의 향상이 예지 선택에서 오는지 연결 방식 정정에서 오는지 확인하기 위해 제안 기법이 선택한 예지 집합에 Union graph의 연결 방식 레이블을 그대로 적용한 중간 구성을 표 6에 함께 제시하였다. 이 구성의 판정 F1은 0.847로 Union graph의 0.854보다 오히려 0.007 낮았으며, 여기에 제안 기법의 연결 방식 레이블을 적용하자 0.864로 0.017 상승하였다. 본 비교 조건에서는 예지 선택만으로 판정 성능이 개선되지 않았으며 연결 방식의 정정이 판정 F1 향상의 주요 요인으로 나타났다. 이는 관계 F1이 Union graph보다 낮음에도 판정 F1이 높은 결과를 설

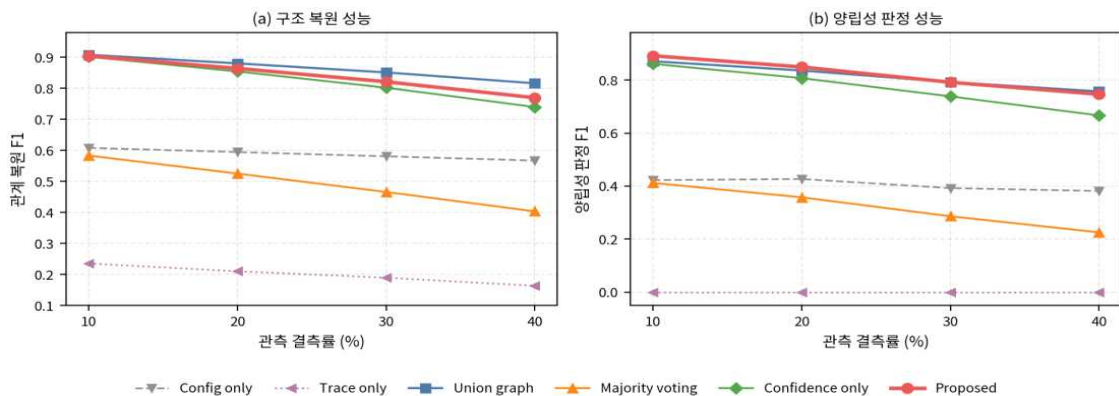


그림 2. 관측 결측률에 따른 성능 변화
Fig. 2. Performance under Varying Observation Missing Rates

명한다.

그림 2는 세 관측 소스에 동일한 결측률을 10%에서 40%까지 적용한 결과이다. 양립성 판정에서 제안 기법은 결측률 10%와 20%에서 각각 0.892와 0.850으로 가장 높았으며 30% 이상에서는 Union graph와의 차이가 0.011 이내로 좁혀졌다. 결측이 심해질수록 서비스별 빌드 방식 사전확률을 추정할 근거가 줄기 때문에 계측 범위가 확보된 환경일수록 제안 기법의 성능이 극대화된다. 구조 정정에서는 Union graph가 모든 구간에서 가장 높은 관계 F1을 유지하였는데 두 지표에서 순위가 다르다는 점은 에지를 많이 포함하는 구조가 반드시 정확한 판정으로 이어지는 않음을 보여준다. 그림 3은 주요 하이퍼파라미터에 대한 민감도를 나타낸다. 후보 수 K 는 25 이상에서 판정 F1이 0.863에서 0.864로 수렴하여 $K=50$ 과 $K=100$ 의 차이가 나타나지 않았으며 $K=10$ 으로 줄여도 0.861로 성능 저하가 0.003에 그쳤다. 주변 판정 임계값 t 는 0.10에서 0.25 사이에서 판정 F1이 0.862에서 0.864로 안정적이었고 0.30을 넘어서면서 재현율이 급격히 낮아져 $t=0.50$ 에서는 0.845로 떨어졌다. 이는 임계값을 0.5보다 낮게 두는 3.4절의 설계가 평가 집합에서도 유효함을 보여준다.

4.4 충돌 해소 추천 평가

충돌 해소 추천은 제안 기법이 충돌을 검출한 288개 실험을 대상으로 평가하였으며 충돌이 검출되지 않은 나머지 12개 실험은 추천 대상에서 제외하였다. 각 실험에서 추천된 1순위 전략을 정답 구조에 적용한 뒤 재판정하여 충돌 서비스 수가 감소하는지를 확인하였다. 그 결과 추천 전략이 생성된 실험은 278개로 검출 대비 96.5%였으며 이 중 221개 실험에서 충돌이 감소하여 전략이 생성된 경우의 성공률은 79.5%, 검출된 전체 실험 기준 성공률은 76.7%였다. 추천된 전략은 구성요소 치환이 225건, 연결 방식 전환이 33건, 노출 경로 차단이 20건이었다. 본 실험에서 생성된 전략은 모두 하나의 에지 또는 정점만을 변경하여 수정 범위가 동일하였으므로 우선순위는 위험 감소량으로 결정되었다. 이 지표는 규칙 집합 R 을 기준으로 한 해소 가능성을 나타내며 전략 적용에 필요한 실제 개발 비용은 반영하지 않는다. 충돌이 감소하지 않은 사례는 제안 기법이 검출한 충돌이 오탐이었거나 선택된 구조에서 산정한 위험 감소량이 정답 구조에서는 실현되지 않은 경우였다.

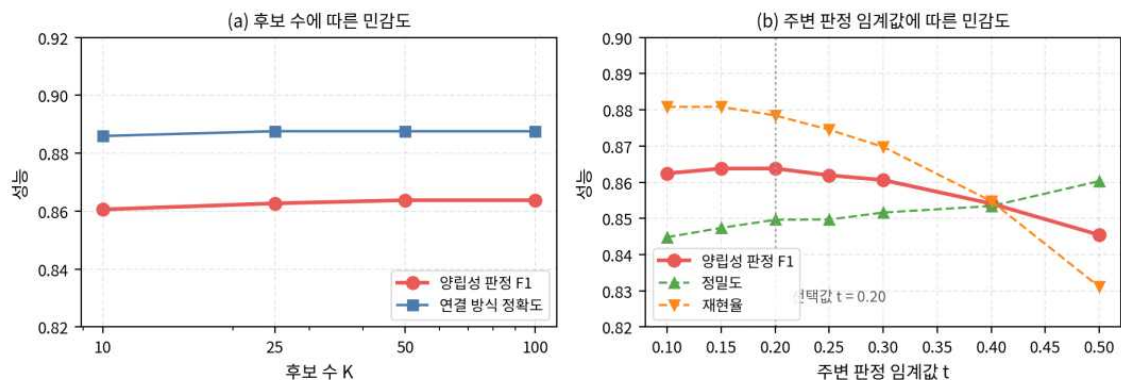


그림 3. 하이퍼파라미터 민감도
Fig. 3. Sensitivity to Hyperparameters

5. 결 론

클라우드 네이티브 환경에서 널리 채택되는 마이크로서비스 아키텍처는 다수의 오픈소스 구성요소를 포함하므로 시스템 내에 다양한 라이선스가 공존하게 된다. 이러한 라이선스 의무의 전과 범위는 구성요소 간의 결합 방식에 따라 결정되나 실제 운영 환경에서 수집되는 설정 정보, 추적 정보, SBOM 등의 관측 데이터는 표본 추출에 따른 누락이나 수집 도구의 편차로 인해 불완전성을 내포한다. 본 논문에서는 이러한 관측 불완전성의 영향을 완화하기 위해 이중의 관측 데이터로부터 복수의 서비스 구조 후보를 확률적으로 생성하고 라이선스 양립성 판정의 주변 분포에 가장 일관된 구조를 최적의 구조로 선택하여 재판정 및 충돌 해소 전략을 추천하는 기법을 제안하였다.

실험을 통한 성능 측정 결과 제안 기법은 서비스 단위 판정 F1 점수 0.864로 다섯 개 기준 비교기법보다 높은 값을 기록하였다. 다만 구조 복잡도 향 제거 구성에서는 0.870을 기록하여 본 평가 조건에서는 해당 향의 추가적인 성능 이득이 확인되지 않았다. 비록 관계 F1은 관측된 모든 예지를 수용한 Union graph가 소폭 높았으나 제안 기법은 연결 방식 정확도(0.888)를 향상시켜 전체 구조의 편집 거리를 13.9% 감소시켰다. 이는 구조 정정과 컴플라이언스 관점의 정확한 판정이 서로 다른 최적화 목표임을 보여주며 제안 기법이 합성 시나리오 환경에서 양립성 판정을 개선할 수 있음을 보여준다.

본 연구는 제안 기법의 성능을 객관적으로 평가하기 위해 결측이 통제된 합성 시나리오를 활용하였으나 실제 운영 시스템의 복잡한 특성을 반영하기 위한 추가적인 확장이 요구된다. 특히 본 실험은 서비스 내부의 빌드 방식 동질성을 85%로 설정한 합성 데이터를 사용하였는데 이는

제안 기법이 활용하는 사전 분포의 가정과 동일한 조건이므로 동질성이 낮은 실제 시스템에서는 연결 방식 정정의 이득이 감소할 수 있다. 또한 라이선스 전과 규칙을 결합 방식과 네트워크 노출 여부 중심으로 단순화하였으므로 개별 라이선스의 세부 조항 해석이 필요한 사례는 본 평가에 포함되지 않는다. 또한 후보 생성 범위가 통합 관측 그래프에 한정되므로 모든 관측원에서 동시에 누락된 참 예지는 본 기법의 정정 대상에 포함되지 않는다. 이에 향후 연구에서는 실제 오픈소스 기반 MSA 벤치마크를 구축하고 배포 산출물 분석 기법을 결합함으로써 관측 결측이 심한 환경에서도 선언 전용 구성요소를 정밀하게 식별하고 시스템의 강건성을 높이는 연구를 수행할 계획이다.

본 연구는 문화체육관광부 및 한국콘텐츠진흥원 부설 한국문화기술기획평가원의 2026년도 SW저작권 생태계 조성 기술개발 사업으로 수행되었음 (과제명 : 클라우드 서비스 활용 구축 형태별 대규모 소프트웨어 라이선스 검증 기술개발, 과제번호 : RS-2023-00224818, 기여율: 100%)

참 고 문 헌

- [1] V. Velepucha, P. Flores, "A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges", IEEE Access, Vol.11, pp.88339-88358, 2023, DOI : 10.1109/ACCESS.2023.3305687
- [2] W. Xu, H. He, K. Gao, M. Zhou, "Understanding and Remediating Open-Source License Incompatibilities in the PyPI Ecosystem", Proceedings of the 38th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp.178-190, 2023,

- DOI : /10.1109/ASE56229.2023.00175
- [3] D.-W. Kim, K.-Y. Park, Y. Joe, D.-M. Shin, "A Study on a License Compatibility Determination Model Using LLM-Based Clause Understanding and Attribute Extraction", *Journal of Software Assessment and Valuation*, Vol.21, No.3, pp.31-46, 2025, DOI : <http://dx.doi.org/10.29056/jsav.2025.09.04>
 - [4] Free Software Foundation, "GNU Affero General Public License, Version 3", 2007, <http://www.gnu.org/licenses/agpl-3.0.html>
 - [5] S. Xu, Y. Gao, L. Fan, Z. Liu, Y. Liu, H. Ji, "LiDetector: License Incompatibility Detection for Open Source Software", *ACM Transactions on Software Engineering and Methodology*, Vol.32, No.1, Article 22, pp.1-28, 2023, DOI : <https://doi.org/10.1145/3518994>
 - [6] D. Kim, S.-J. Cho, S. Han, M. Park, and I. You, "Open Source Software Detection using Function-level Static Software Birthmark," *Journal of Internet Services and Information Security*, vol. 4, no. 4, pp. 25 - 37, 2014, doi: 10.22667/JISIS.2014.11.31.025.
 - [7] National Telecommunications and Information Administration, "The Minimum Elements for a Software Bill of Materials (SBOM)," U.S. Department of Commerce, 2021. <https://www.ntia.gov/report/2021/minimum-elements-software-bill-materials-sbom>
 - [8] SPDX Workgroup, "The Software Package Data Exchange (SPDX) Specification, Version 2.3", The Linux Foundation, 2022, <https://spdx.github.io/spdx-spec/v2.3/>
 - [9] OWASP Foundation, "CycloneDX Bill of Materials Specification", <https://cyclonedx.org/specification/overview/>
 - [10] B. Xia, T. Bi, Z. Xing, Q. Lu, L. Zhu, "An Empirical Study on Software Bill of Materials: Where We Stand and the Road Ahead", *Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE)*, pp.2630-2642, 2023, DOI : <https://doi.org/10.1109/ICSE48619.2023.00219>
 - [11] Y. Joe, H.-S. Kim, D.-M. Shin, "Pseudo Open Source Licenses and License Compatibility Checking", *Journal of Software Assessment and Valuation*, Vol.20, No.4, pp.55-62, 2024, DOI : /10.29056/jsav.2024.12.06
 - [12] OpenTelemetry Authors, "OpenTelemetry Specification", Cloud Native Computing Foundation, <https://opentelemetry.io/docs/specs/otel/>
 - [13] S. Ashok, V. Harsh, B. Godfrey, R. Mittal, S. Parthasarathy, L. Schwartz, "TraceWeaver: Distributed Request Tracing for Microservices Without Application Modification", *Proceedings of the ACM SIGCOMM 2024 Conference*, pp.828-842, 2024, DOI : <https://doi.org/10.1145/3651890.3672254>
 - [14] S. Wang, H. Zhang, Q. Z. Sheng, X. Li, Z. Sun, T. Cai, W. Zhang, J. Yang, Q. Gao, "A Survey on Truth Discovery: Concepts, Methods, Applications, and Opportunities", *IEEE Transactions on Big Data*, Vol.11, No. 2, pp.314-332, 2025, DOI : <https://doi.org/10.1109/TBDATA.2024.3423677>
 - [15] F. Azzalini, D. Piantella, E. Rabosio, L. Tanca, "Enhancing domain-aware multi-truth data fusion using copy-based source authority and value similarity", *The VLDB Journal*, Vol.32, No.3, pp.475-500, 2023, DOI : <https://doi.org/10.1007/s00778-022-00757-x>

저 자 소 개



박경엽(Kyung-Yeob Park)

2019.2 서울과학기술대학교
컴퓨터공학과 석사
2019-현재
<주관심분야> 엘에스웨어(주) 책임 연구원
IoT 보안, 블록체인, 빅데이터,
메타버스



김동완(Dong-Wan Kim)

2022.02 한국성서대학교
컴퓨터소프트웨어학과 졸업
2024.02 숭실대학교 컴퓨터학과 석사
2024-현재
<주관심분야> 엘에스웨어(주) 주임 연구원
클라우드, 인공지능, 빅데이터,
블록체인



조용준(YongJoon Joe)

2011.3 큐슈대학교
전기정보공학과 졸업
2013.3 큐슈대학교 정보학부 석사
2016.3 큐슈대학교 정보학부
박사과정 수료
2013.4-2016.3 일본 학술진흥원 특별연구원
2016-현재
<주관심분야> 엘에스웨어(주) 이사
병렬·분산 컴퓨팅, 게임이론,
분산 제약 최적화 문제



신동명(Dong-Myung Shin)

2003.02 대전대학교
컴퓨터공학과 박사
2001-2006 한국정보보호진흥원
응용기술팀 선임연구원
2006-2014 한국저작권위원회
저작권기술팀 팀장
2014-2016 한국스마트그리드사업단보안
인증팀 팀장
2016-현재 엘에스웨어(주)
소프트웨어연구소 연구소장/
전무이사
<주관심분야> 오픈소스 라이선스, 저작권 기
술, 시스템/네트워크보안, SW
취약점 분석·감정, 블록체인
기술, 메타버스