

# 이상탐지(Anomaly Detection)

## 결과보고서

작성일: 2025년 12월 21일

작성자: 송공호(2025254010)

주제: 머신러닝 기반 이상탐지 모델 개발 및 성과 평가

## 1. 개요

### 1.1 목적

제조 공정 및 시스템 모니터링에서 정상 상태에서 크게 벗어난 비정상 데이터(이상, Anomaly)를 조기에 탐지하기 위한 머신러닝 기반 솔루션 개발 및 평가

### 1.2 이상탐지의 정의 및 필요성

이상탐지는 일반적인 데이터와는 다른 메커니즘에 의해 발생된 데이터를 식별하는 프로세스입니다[1].

#### 주요 특징:

- 정상(Normal)과 비정상(Abnormal)을 구분하는 분류 문제
- 비정상 데이터의 심각한 부족으로 비지도 학습에 가까운 특성 보유
- 정상 데이터의 특성을 학습하여 정상 범위 추정
- 제조 공정: 불량 제품 조기 탐지
- 금융: 비정상 거래 및 사기 탐지
- 보안: 침입 및 비정상 접근 탐지

## 2. 이상탐지의 도전 과제 및 해결 방안

### 2.1 핵심 도전 과제[1]

#### ① Label 확보의 어려움

- **문제:** 정상 데이터는 자동 수집되지만, 비정상 데이터는 대부분 수동 관리
- **영향:** 신뢰성 있는 모델 평가를 위한 레이블 부족
- **사례:** 선박 엔진 부품 결함 예측 프로젝트 (레이블 전무), 제강라인 불량 예측 (레이블 신뢰도 문제)

## ② 낮은 성능

- 이상탐지 모델은 일반적으로 높지 않은 성능 제시
- 그래프 기반 평가시 모델 사용 가능성 판단 필요

## ③ 올바른 평가 지표 선정

- 단순 정확도(Accuracy)는 부적절 (다수 클래스 편향)
- 재현율(Recall)과 정밀도(Precision) 간 균형 필요
- Cut-off Value 결정의 모호함 (회색 지대 존재)

# 2.2 해결 방안

## A. Class Imbalance 문제 해결[1]

**클래스 불균형이 발생하는 상황:**

- 고객 이탈 예측: 잔존 >> 이탈
- 금융 비정상 거래: 정상 >> 비정상(사기)
- 제조 공정 불량: 정상 >> 불량

**문제점:**

결과적으로 높은 정확도에도 불구하고 소수 클래스(비정상)의 재현율이 매우 낮음

**해결책:**

1. **Resampling 방식:** Up-sampling, Down-sampling, SMOTE
2. **Class Weight 조정:** 손실 함수(Loss Function)에서 소수 클래스에 더 높은 가중치 부여[1]

방법	설명
<code>class_weight = 'None'</code>	기본값, 가중치 미적용
<code>class_weight = 'balanced'</code>	<code>y_train</code> 의 클래스 비율을 역으로 적용
<code>class_weight = {0:w0, 1:w1}</code>	비율 지정 (합 = 1)

Table 1: Class Weight 조정 옵션

## B. 올바른 평가 지표: F1-Score

F1-Score는 정밀도(Precision)와 재현율(Recall)의 조화평균으로, 클래스 불균형 상황에서 소수 클래스의 성능을 균형있게 평가[1]:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

특징:

- 재현율과 정밀도를 동시에 고려
  - 단순 산술평균이 아닌 조화평균 사용
  - Cut-off value와 무관한 평가 지표: ROC Curve, Precision-Recall Curve
- 

## 3. 머신러닝 기반 이상탐지 모델

### 3.1 Isolation Forest[1]

개념

데이터를 고립(Isolation)시키기 위한 트리 개수와 깊이(Depth)를 이용하여 정상/비정상 판별

알고리즘 동작 원리

#### Step 1: 학습 데이터 샘플링

- 원본 학습 데이터에서 256개 정도 샘플 추출 (경험적 권장값)

#### Step 2: Isolation Tree 생성

- 랜덤하게 feature 선택
- 랜덤하게 split 기준값 설정
- 모든 데이터가 leaf node에 고립될 때까지 반복
  - 정상 데이터: 고립에 많은 분할 필요 → Depth 증가
  - 비정상 데이터: 쉽게 고립 → Depth 감소

Step 3: 이상 점수(Anomaly Score) 계산

각 데이터포인트의 평균 Depth를 기반으로 정상/비정상 점수 계산:

$$s(x_i,n) = 2^{-\frac{E(h(x_i))}{c(n)}}$$

여기서:

- $E(h(x_i))$ : 개별 데이터의 평균 경로 깊이
- $c(n)$ : 정상화 상수
- $s(x_i,n)$ : 1에 가까울수록 비정상

하이퍼파라미터

파라미터	설명	기본값
n_estimators	생성하는 Isolation Tree 개수	100
max_samples	각 Tree 학습에 사용하는 샘플 수	256
contamination	데이터에서 비정상으로 간주할 비율	자동
max_depth	최대 깊이 (log <sub>2</sub> n으로 고정)	8 (n=256)

Table 2: Isolation Forest 하이퍼파라미터

모델 튜닝 방법

1. Contamination 조정
  - 학습 데이터의 비정상 비율에서 시작
  - 검증 데이터에서 성능을 평가하며 조정
  - F1-Score 최대화를 목표
2. Cut-off Value 최적화
  - 검증 데이터로 이상 점수 계산
  - Cut-off를 조정하며 F1-Score 추적
  - F1 최대값을 제공하는 Cut-off 선택
3. 하이퍼파라미터 튜닝
  - n\_estimators 조정 (50 ~ 200)

- max\_depth 조정
- for 루프를 통한 격자 탐색(Grid Search)

## 장점과 단점

### 장점:

- 비지도 학습으로 레이블 의존성 낮음
- 계산 효율성 우수
- 고차원 데이터에서도 성능 유지
- 이상 점수 직관적 해석

### 단점:

- Context 기반 이상탐지 어려움 (시계열 데이터)
- Collective anomaly 탐지 불가
- 낮은 contamination 값에서 성능 저하

---

## 4. 딥러닝 기반 이상탐지 모델

### 4.1 AutoEncoder[1]

#### 개념

정상 데이터로만 학습하여 정상의 특징을 압축 표현으로 학습한 후, 입력 데이터를 원래대로 복원하는 과정에서의 오차를 이용

#### 모델 구조

##### AutoEncoder 구조

Figure 1: AutoEncoder 예시 구조: 52차원 입력

#### 동작 원리

##### 학습 단계 (Training):

1. 정상(Normal) 데이터만으로 학습
2. Encoder: 입력 데이터를 저차원 표현으로 압축

3. Decoder: 압축된 표현을 원래 차원으로 복원
4. 손실 함수(MSE)를 통해 입력과 복원 오차 최소화
5. 최적화 과정에서 정상 데이터의 중요한 특징만 보존

#### 복원 오차(Reconstruction Error) 계산:

$$RE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

여기서:

- $x_i$ : 원본 입력값
- $\hat{x}_i$ : 복원된 값
- $n$ : 특성(변수) 개수

#### 검증 단계 (Validation):

1. 정상과 비정상 모두 포함된 검증 데이터 사용
2. 복원 오차 계산
  - 정상 데이터: 오차가 작음 (학습한 패턴)
  - 비정상 데이터: 오차가 큼 (미학습 패턴)
3. Threshold 기반 분류
  - 오차 > Threshold → 비정상
  - 오차 ≤ Threshold → 정상

## 모델 성능 최적화

#### 전처리:

- MinMax Scaling 적용 (범위: 0~1)
- 정상 데이터만 추출: `x_train0 = x_train[y_train == 0]`

#### 학습 설정:

- 손실 함수: Mean Squared Error (MSE)
- 최적화 알고리즘: Adam
- 에포크: 50, 배치 크기: 64
- 검증 데이터 분할: 20%

#### Threshold 결정 절차:

1. 재구성 오차 분포 시각화

- 정상 vs 비정상 데이터의 오차 분포 확인
- 겹치는 영역(회색지대) 파악

## 2. Precision-Recall-F1 곡선 그리기

- Threshold를 조정하며 정밀도, 재현율, F1-Score 추적
- 그래프 상에서 세 지표의 변화 관찰

## 3. 최적 Threshold 선택

- F1-Score 최대값을 제공하는 Threshold 선택
- 또는 비즈니스 요구사항에 따라 수동 조정

## 4. 최종 성능 평가

- 선택된 Threshold로 분류
- Confusion Matrix, Classification Report 생성

## 장점과 단점

### 장점:

- 고도의 비선형 패턴 학습 가능
- 자동 특성 추출 (Feature Engineering 불필요)
- 대용량 고차원 데이터 처리 능력
- 시계열 이상탐지 가능 (LSTM-AE, CNN-AE)

### 단점:

- 학습 데이터에 대한 의존성 높음
- 과적합(Overfitting) 위험
- 해석 어려움 (Black Box)
- 계산 비용 높음

---

# 5. 비즈니스 관점의 성능 평가

## 5.1 기술적 평가 vs 비즈니스 평가

평가 관점	기술적 평가	비즈니스 평가
초점	정확도 지표	비즈니스 임팩트

목표	성능 최대화	가치 최대화
고려사항	F1-Score, AUC	비용, 수익, 위험
결정 주체	데이터 사이언티스트	경영진

Table 3: 평가 관점별 차이

## 5.2 혼동행렬(Confusion Matrix) 기반 분석

	실제 정상	실제 비정상
예측 정상	TN (정상 올바르게)	FN (비정상 놓침)
예측 비정상	FP (오경보)	TP (비정상 탐지)

Table 4: 혼동행렬 구조

### 비즈니스 관점 해석:

- **False Negative (FN):** 실제 불량품을 정상으로 판단
  - 심각한 비용 발생 (고객 불만, 리콜, 브랜드 손상)
  - 최소화 필요 (높은 Recall)
- **False Positive (FP):** 정상 제품을 불량으로 판단
  - 불필요한 재검사 비용
  - 생산 라인 중단
  - 수용 가능한 수준의 Trade-off

## 5.3 모델 선택 기준

1. **Recall 우선** (암 진단 같은 중대 상황)
  - False Negative 최소화 중시
  - "놓치면 안 되는" 비정상 탐지
  - 일부 False Positive 수용
2. **Precision 우선** (스팸 필터 같은 상황)
  - False Positive 최소화 중시

- "실제 정상을 비정상으로 판단"하는 비용 높음
- 일부 False Negative 수용

### 3. F1-Score 균형 (일반적 상황)

- Recall과 Precision의 조화평균
  - 양쪽 모두 중요한 상황
- 

## 6. 모델 적용 절차: CRISP-DM

이상탐지 프로젝트는 다음의 반복적 프로세스를 따릅니다[1]:

### 1. Business Understanding

- 비즈니스 문제 정의
- 데이터 분석 방향 및 목표 설정
- 초기 가설 수립

### 2. Data Understanding

- 원본 데이터 식별
- 분석 구조 설계
- EDA (Exploratory Data Analysis) 및 CDA (Contextual Data Analysis)

### 3. Data Preparation

- 모든 셀에 값이 있어야 함
- 모든 값은 숫자여야 함
- 필요시 숫자 범위 일치화

### 4. Modeling

- 모델 구축 및 검증
- 기술적 관점 평가
- 비즈니스 관점 평가

### 5. Deployment

- 모델 관리 및 모니터링
- AI 서비스 구축

## 6. Evaluation

- "무엇이 문제인가?"
- "문제가 해결되었는가?"
- 필요시 1단계로 돌아가 반복

---

# 7. 적용 사례: 반도체 제조 공정[1]

## 7.1 프로젝트 개요

- **목표:** 반도체 제조 공정의 불량품 조기 탐지
- **데이터:** 공정 중 수집된 센서 데이터
- **레이블:** 불량 여부 (정상 vs 불량)

## 7.2 데이터 특성

- 클래스 불균형: 정상 >> 불량 (예: 95% 정상, 5% 불량)
- 고차원 데이터: 50+ 센서 변수
- 시계열 특성: 시점별 센서 값 변화

## 7.3 모델 적용 결과

- **기본 모델** (Class Weight 미적용): 높은 정확도, 낮은 불량 탐지율
- **개선 모델** (Class Weight 적용): 불량 탐지율 향상, 전체 성능 균형 개선

---

# 8. 결론 및 추천

## 8.1 주요 결론

1. **Class Imbalance는 필수 고려사항**
  - 클래스 불균형 상황에서 기본 모델은 소수 클래스 성능 급락
  - Class Weight 조정 또는 Resampling 필수
2. **올바른 평가 지표 필수**
  - 단순 정확도는 오도 가능

- F1-Score, Precision-Recall Curve 기반 평가 권장

### 3. 모델 선택의 유연성

- Isolation Forest: 빠르고 간단한 구현 (머신러닝)
- AutoEncoder: 복잡한 패턴 학습 (딥러닝)
- 데이터 특성과 요구사항에 따라 선택

### 4. Threshold 결정은 비즈니스와의 협의 필수

- 기술적 최적값(F1 최대)과 비즈니스 요구 간 조율 필요
- False Negative vs False Positive의 비용 고려

## 8.2 추천 사항

#### 1. 단기 (Phase 1)

- Isolation Forest 파일럿 프로젝트 추진
- 데이터 품질 및 레이블 신뢰도 확보

#### 2. 중기 (Phase 2)

- AutoEncoder 기반 모델 평가
- LSTM-AE 또는 CNN-AE 등 고급 모델 탐색

#### 3. 장기 (Phase 3)

- Ensemble 모델 구축 (IF + AE 결합)
- 실시간 모니터링 시스템 구축
- 지속적인 모델 성능 모니터링 및 재학습

## 8.3 주의사항

- 정기적인 모델 성능 모니터링 필수
- 새로운 비정상 패턴 학습을 위한 재학습 절차 수립
- 모델 해석 가능성과 투명성 확보
- 이해관계자(경영진, 운영팀) 소통 및 신뢰 구축

---

## 참고문헌

[1] 한기영 (2024). *이상탐지(Anomaly Detection)*. 데이터인사이트.

[2] scikit-learn Documentation. Isolation Forest. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html>

[3] Zhou, Z. H., & Liu, X. Y. (2008). Training Cost-sensitive Deep Classifiers with Winequality Dataset. *International Conference on Data Mining (ICDM)*.

[4] Keras Documentation. Autoencoder. <https://keras.io/>

---

## 문서 정보

- 버전: 1.0
- 작성일: 2025-12-28
- 최종 검토일: 2025-12-28
- 담당부서: R&D 분석팀