

# 시계열 특성을 활용한 다양한 이상 거래 탐지 딥러닝 모델 개발과 성능 분석



지도교수: 송길태

분과: 소프트웨어/인공지능 (A)

팀명: 송골매

팀원: 배근호(202055551)

# 목차

1. 요구조건 및 제약 사항 분석에 대한 수정사항 .....	3
1.1    요구 조건 수정 .....	3
1.2    제약 사항 및 극복 방안 .....	4
2. 설계 상세화 및 변경 내역 .....	5
2.1    개발 언어 및 개발 도구 .....	5
2.2    시스템 구성도 및 과정 .....	6
3. 갱신된 과제 추진 계획 .....	8
4. 구성원별 진척도 .....	8
5. 보고 시점까지의 과제 수행 내용 및 중간 결과 .....	9

## 2025 전기 졸업과제 중간보고서

### 1. 요구조건 및 제약 사항 분석에 대한 수정사항

#### 1.1 요구 조건 수정

기존에는 거래, 상점, 고객, 카드 등 다양한 실체를 노드로 설정하고 PyTorch Geometric 기반의 GCN, GAT 등 그래프 신경망 모델을 설계하여 동일 고객 또는 카드의 연속 거래를 시계열 방향성(edge)으로 연결하는 방식을 계획하였다. 하지만 이러한 구조는 거래의 복잡한 시계열 정보와 방향성이 모델 내에서 충분히 반영되지 않는다는 한계가 예상된다.

#### 개선된 탐지 모델 설계 방향

이러한 한계를 극복하기 위해, 이번 과제는 다음과 같은 방향으로 탐지 모델을 재설계하고자 한다.

- **트랜스포머(Transformer) 활용 시계열 특성 추출**

거래의 시계열 데이터를 Transformer 기반 딥러닝 모델로 우선적으로 처리하여, 시계열적 패턴과 거래의 시간적 종속성을 효과적으로 임베딩(embedding)으로 추출한다.

- **거래 임베딩 + 고객/카드 데이터 통합**

Transformer에서 얻은 거래 임베딩에 고객 및 카드 관련 특성을 결합해 최종 입력 feature를 구성한다.

- **세 가지 이상 거래 탐지 모델 적용 및 성능 비교**

1. **TGN (Temporal Graph Network):** 시간의 흐름과 그래프 구조 모두를 반영하는 모델로, 거래 데이터의 시계열성과 엔터티 간 관계를 통합해 이상 거래 탐지
2. **Heterogeneous Graph Neural Network:** 고객/카드/상점 등 다양한 엔터티와 관계 정보를 이질적인 그래프로 모델링하여, 다층적 상호작용 파악
3. **MLP (Multi-Layer Perceptron):** 시계열+비정형 데이터 기반 간단한 딥러닝 모델로, 비교를 위한 베이스라인 제공

#### 성능평가 및 적용 기준

- **평가 지표**

- Precision, Recall, F1-score, ROC-AUC 등 다양한 분류 성능 지표 활용

## 2025 전기 졸업과제 중간보고서

- 실제 금융 거래 데이터의 심각한 불균형 상황을 고려해, F1-score와 Recall을 핵심 평가 기준으로 설정

### • 예측 결과 예시 및 평가 기준

- 각 거래별로 이상 거래 점수(0~1 범위)를 산출하며, 실제 라벨과 비교하여 정밀도(Precision), 재현율(Recall) 및 F1-score 측면에서 모델 평가

### • 적용 범위 구체화

- 개발된 이상 거래 탐지 모델은 금융기관, 카드사, PG사 등 실거래 및 결제 데이터 기반 리스크 관리 업무에 직접 적용 가능
- 또한 이커머스, 온라인 결제 등 대규모 실서비스 환경에서의 실시간 이상 거래 탐지에도 확장 적용 가능

## 1.2 제약 사항 및 극복 방안

이상 거래 탐지는 정상 거래 대비 발생 빈도가 극히 낮아, 확보한 데이터에서도 정상 거래와 이상 거래의 비율이 99.85:0.15로 극심한 불균형을 보인다. 착수보고서에 대한 자문의견서의 내용을 반영하여 불균형 문제를 극복하기 위해 사기 거래 데이터는 증강시키고 정상 거래 데이터는 언더샘플링하여 한계점을 보완한다.

시계열 구조를 보존하면서 데이터를 생성할 수 있는 윈도우 슬라이싱 기법과 노이즈 적용 방법을 사용해 사기 거래가 갖는 시간 흐름의 특징(시간대, 패턴)과 속성간 동일 조건(우편번호 같으면 주, 도시 정보 같음)을 유지한 synthetic sample 생성해 사기 거래 데이터를 증강시키고 정상 거래에 대해 랜덤으로 언더샘플링하기 보다 사기 거래와 유사한 분포를 가진 정상 거래를 우선 샘플링하고 부족하면 추가로 샘플링하여 모델이 유의미한 패턴에 집중할 수 있도록 하여 데이터 불균형 문제를 해소한다.

## 2. 설계 상세화 및 변경 내역

### 2.1 개발 언어 및 개발 도구

#### 2.1.1 개발 언어

- DBMS: PostgreSQL
- 모델 개발: Python

#### 2.1.2 모델 개발 관련 라이브러리

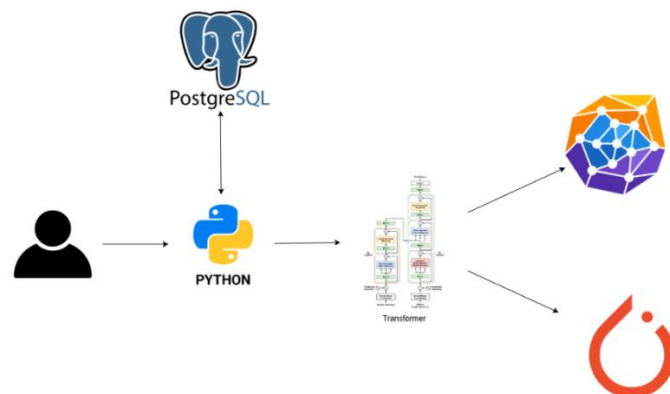
- pandas
  - 데이터 프레임 구조를 이용한 데이터 로딩, 전처리, 조작 등에 사용
  - 모델 입력을 위한 데이터 가공 및 통합 작업 수행
- numpy
  - 수치 계산 및 배열 처리를 위한 라이브러리
- scikit-learn (sklearn)
  - 머신러닝 모델 구축과 평가, 데이터 분할, 전처리 등을 위한 라이브러리
  - 주요 사용 모듈:
    - train\_test\_split: 학습/검증 데이터 분리
    - LabelEncoder: 범주형 변수 인코딩
    - StandardScaler: 데이터 정규화
    - RandomForestClassifier: 트리 기반 분류 모델 구축, feature\_importances\_로 중요 속성 추출
    - NearestNeighbors: 정상 거래 언더샘플링 과정에서 사기 거래에 가까운 정상 거래들을 선택하기 위해 사용
- xgboost
  - 이상 거래 탐지와 같은 불균형 데이터 문제에 강건한 분류 모델 학습
  - shap 라이브러리로 중요 속성 추출을 위해 사용
- shap (SHapley Additive exPlanations)
  - 머신러닝 모델의 예측 결과에 대한 설명력 확보를 위해 사용

## 2025 전기 졸업과제 중간보고서

- 모델의 개별 속성들이 예측에 어떤 영향을 미쳤는지 시각화 및 정량적으로 분석
- autofeat (AutoFeatClassifier)
  - 자동으로 속성 생성 및 선택을 지원하는 라이브러리
  - 수학적 변환 기반의 복합 파생변수를 생성하여 모델 성능 개선을 도모
- geopy
  - 거래 데이터 내 zip(우편번호), 미국의 경우 주(State) 및 도시(City), 해외 거래의 경우 국가명과 도시명 정보를 활용하여 해당 거래의 위도(latitude)와 경도(longitude) 좌표값을 조회하는 데 사용
  - 산출된 위도/경도 좌표는 거래 간 위치 변화를 구체적으로 수치화할 수 있게 하며, 이후 각 거래 쌍 사이의 발생 시간 차이와 이동 거리(위치 변화)를 계산하여 이상 거래 탐지 모델의 시계열적 및 공간적 분석에 활용
  - 짧은 시간 내 먼 거리에서 연속 발생하는 거래 등 공간적 이상 패턴 검출에 중요한 입력 변수로 작용
- PyTorch
  - Python 기반으로 개발된 대표적인 오픈소스 딥러닝 프레임워크
  - MLP(다층 퍼셉트론 등) 및 각종 기본 신경망 구현
- PyTorch Geometric (PyG)
  - PyTorch 기반의 그래프 신경망(GNN) 전용 라이브러리
  - 그래프 기반 데이터 처리 및 다양한 GNN 모델 구현에 최적화

## 2.2 시스템 구성도 및 과정

### 2.2.1 시스템 구성도



## 2025 전기 졸업과제 중간보고서

### 2.2.2 시스템 개발 과정

- 데이터 전처리:
  - 거래, 고객, 카드 데이터에서 개별, 연속 거래 시퀀스 생성
  - 거래 간 시간/위치 간격 등 주요 시계열 정보 산출
  - 이상 탐지에 효과적인 파생변수, 전처리 특성도 함께 추출
- 시계열 패턴 추출 (Transformer 활용)
  - 각 고객 또는 카드 단위 거래 시퀀스를 Transformer 기반 시계열 신경망에 입력
  - 거래 패턴, 시계열적 순서, 연속성 등 복합적인 고차원 임베딩 벡터로 변환
  - 생성된 임베딩은 후속 GNN 및 MLP 모델에 공통 입력
- 그래프 데이터 구성
  - 고객, 카드, 거래 등 엔티티를 그래프의 노드로 설정
  - 실제 관계(동일 고객·카드 기반 거래, 연속성 등)는 엣지로 연결
  - 거래의 시계열 특성을 반영하여 방향성(Directed Edge) 정보 포함
- 3가지 딥러닝 모델 구축 및 학습
  - TGN (Temporal Graph Network)
    - 시계열 방향성 그래프의 상호작용을 학습
    - 노드/엣지의 시간·관계 변화를 동시에 반영
  - 이종 그래프 신경망 (Heterogeneous Graph)
    - 고객, 카드, 거래, 상점 등 다양한 유형의 엔티티·관계를 정교하게 모델링
  - MLP (Multi-Layer Perceptron)
    - 시계열 임베딩과 주요 거래 특성값을 입력으로 단순한 신경망 학습
- 지도학습 및 예측
  - 각 모델별로 정상/사기 라벨이 포함된 데이터를 통한 지도학습 수행
  - 거래 특성, 시계열 임베딩 벡터를 입력으로 사기 가능성 확률(0~1)을 출력
- 추론 및 성능 평가
  - F1-score, Recall, Precision, ROC-AUC 등 다양한 분류 지표로 성능 평가
  - 불균형 데이터 특성을 고려해 F1-score 및 Recall을 핵심 비교 기준으로 삼아 3가지 모델의 성능을 분석

## 2025 전기 졸업과제 중간보고서

### 3. 갱신된 과제 추진 계획

업무	6				7				8				9			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
데이터 수집 및 전처리																
모델 개발 지식 학습																
트랜스포머 시계열 데이터 처리 모델																
중간보고서 작성																
이상 거래 탐지 딥러닝 모델 개발																
모델 성능 평가 및 개선																
최종보고서 작성																

### 4. 구성원별 진척도

조원	진척도
배근호	데이터 전처리 <ul style="list-style-type: none"> <li>- 이상 데이터 처리</li> <li>- 데이터 정제 및 파생 데이터 속성 생성</li> <li>- 데이터 중요 속성 분류</li> <li>- 불균형 문제 극복을 위해 데이터 증강 및 언더샘플링</li> </ul>
추민	자료 조사 <ul style="list-style-type: none"> <li>- 금융 데이터 도메인 지식 학습</li> <li>- 다른 이상 거래 탐지 연구 자료에 사용한 속성 조사</li> </ul>
윤소현	데이터 전처리 <ul style="list-style-type: none"> <li>- zip(우편번호) -&gt; 위도, 경도 좌표 변환</li> <li>- 각 거래별 발생 위치 변화, 고객 주소와의 거리 차이 계산</li> </ul>



## 2025 전기 졸업과제 중간보고서

### 5. 보고 시점까지의 과제 수행 내용 및 중간 결과

#### 5.1 데이터 결측치 조사 및 처리

```
## 1. 각 열의 빈 값 개수:  
merchant_state    1563700  
zip                1652706  
errors             13094522  
fraud              4390952  
dtype: int64
```

- **fraud 라벨 결측:**

수집된 거래 데이터 전체 1,300만 건 중 fraud 라벨이 없는 데이터는 전량 삭제하였으며, 그 결과 최종적으로 약 900만 건의 데이터가 남았다. 이는 지도학습 및 이상 탐지 모델의 품질을 높이기 위한 조치이다.

- **errors 속성 결측:**

errors 컬럼의 결측치는 "No Error"로 일괄 대체했으며, 이후 인코딩 처리하여 학습과 분석에 활용할 수 있도록 하였다. 이는 카테고리 결측치 처리의 일반적인 방식으로 정보 손실을 최소화한다.

- **zip 및 merchant\_state 결측:**

두 컬럼의 결측 건수는 거의 일치하였으며, 이는 두 속성이 미국 내 거래의 우편번호와 주(State) 정보를 나타내기 때문이다.

- 온라인 결제(merchant\_city=ONLINE)의 경우, zip과 merchant\_state 모두 미가입되어 있음이 확인되었다.
- 해외 거래의 경우에는 zip은 결측이나, merchant\_state에는 해당 국가명이 입력되어 있다.
- 이후 zip 컬럼은 지역 기반 군집화 및 시계열적 위치 분석 등에서 활용할 가능성이 있어, 해외 거래의 경우 merchant\_state 값(국가명)의 대표 알파벳 앞 두 글자에 '00'을 붙인 5자리 문자열 형식으로 zip 값을 보완 처리하였다.
  - 예: merchant\_state = "FRANCE" → zip = "FR00"

### 5.2 데이터 중요 특징 분류

#### 5.2.1 거래 데이터 중심으로 병합

- 수집된 거래, 고객, 카드 데이터를 거래 데이터의 고객ID와 카드ID를 기준으로 병합하여, 거래를 중심으로 고객·카드의 속성 정보가 결합된 통합 데이터셋을 구성하였다.
- 병합 결과 데이터는 총 35개의 속성을 포함한다.  
(id, date, client\_id, card\_id, amount, use\_chip, merchant\_id, merchant\_city, merchant\_state, zip, mcc, errors, mcc\_type, fraud, current\_age, retirement\_age, birth\_year, birth\_month, gender, address, latitude, longitude, per\_capita\_income, yearly\_income, total\_debt, credit\_score, num\_credit\_cards, card\_brand, card\_type, expires, has\_chip, num\_cards\_issued, credit\_limit, acct\_open\_date, year\_pin\_last\_changed)

#### 5.2.2 파생 속성 생성

```
df['expires_last_day'] = df['expires'].apply(convert_expires_to_last_day)

# 파생 변수 생성
df['hour'] = df['date'].dt.hour
df['dayofweek'] = df['date'].dt.dayofweek
df['account_age_days'] = (df['date'] - df['acct_open_date']).dt.days
df['months_to_expiry'] = (df['expires_last_day'].dt.year - df['date'].dt.year) * 12 + (df['expires_la
df['transaction_age'] = df['date'].dt.year - df['birth_year']
df['years_to_retirement'] = df['retirement_age'] - df['transaction_age']
df['zip_prefix'] = df['zip'].astype(str).str[:3]
```

- 기존의 단일 변수만으로는 설명력이 부족하거나, 값의 종류(카테고리)가 지나치게 많은 변수들의 정보를 요약·확장하기 위해, 거래·고객·카드 정보에서 다양한 파생 속성을 추가로 생성하였다.
- 생성된 주요 파생 변수의 예시는 다음과 같다.
  - hour: 거래 발생 시각. 사기 거래가 특정 시간에 집중되는 경향성을 파악하기 위해 생성
  - dayofweek: 거래 발생 요일. 요일별로 사기 거래 비율 차이를 분석하기 위함
  - account\_age\_days: 계좌(또는 카드) 개설일로부터 거래일까지의 경과 일수. 개설 초기 또는 오랜 계좌 유지 등 시점에 따라 사기 패턴이 달라질 가능성 반영

## 2025 전기 졸업과제 중간보고서

- months\_to\_expiry: 거래 시점 기준 카드의 만료까지 남은 개월 수. 만기가 가까운 카드의 사기 거래 위험성 등 연관 분석 목적
- transaction\_age: 거래 시점 기준 고객의 실질적 나이. 고객별 금융 습관 및 연령대별 특징 반영
- years\_to\_retirement: 해당 거래 시점에서 은퇴까지 남은 예상 연수. 생애 주기 변화에 따른 거래 이상 신호 포착에 참고
- zip\_prefix: 우편번호 앞 3자리 그룹핑(군집화). 지역별, 인접 도시/군 단위 등 중간 규모의 지리 정보 요약

### 5.2.3 중요 속성 분류

- RandomForest의 feature\_importances\_

```
clf = RandomForestClassifier(n_estimators=100, random_state=42)
clf.fit(X_train, y_train)
importances = clf.feature_importances_
feature_names = X.columns
```

feature importance 값은 전체 예측 성능(Gain Info)에 대한 각 변수의 상대적인 기여도 높은 중요도 점수 ⇒ 모델이 정상/사기 거래를 식별할 때 해당 변수에 더 많이 의존

RF 분석 결과 중요도 상위 속성
zip_prefix
mcc_type
merchant_city
amount
merchant_state
use_chip

RF 분석 결과 중요도 하위 속성
has_chip
gender
num_cards_issued
errors
card_brand
card_type

## 2025 전기 졸업과제 중간보고서

- XGBoost + SHAP

```
# XGBoost 모델 훈련
model = xgb.XGBClassifier(
    n_estimators=100, max_depth=6, learning_rate=0.1,
    use_label_encoder=False, eval_metric='logloss'
)
model.fit(X_train, y_train)

# SHAP 분석
explainer = shap.Explainer(model)
shap_values = explainer(X_test)

# SHAP 중요도 수치 출력
mean_abs_shap = np.abs(shap_values.values).mean(axis=0)
shap_importance = pd.DataFrame({
    'feature': X.columns,
    'mean_abs_shap': mean_abs_shap
}).sort_values(by='mean_abs_shap', ascending=False)
```

SHAP 값은 모델 예측값에 대한 각 feature의 기여도(영향력)를 정량적으로 분석

mean\_abs\_shap 값은 각 변수의 SHAP 값 절대값을 평균 낸 것으로, 모든 샘플에서 해당 변수의 평균적인 영향력 크기를 의미

XGBoost + SHAP 분석 결과 중요도 상위 속성
zip_prefix
merchant_city
mcc_type
hour
amount
merchant_state

XGBoost + SHAP 분석 결과 중요도 하위 속성
gender
card_brand
num_cards_issued
year_pin_last_changed
years_to_retirement
errors

- 위 두 모델(Random Forest, XGBoost+SHAP)에서 공통적으로 중요도가 낮게 평가된 gender, card\_brand, num\_cards\_issued, errors, years\_to\_retirement, transaction\_age, has\_chip, credit\_score 등의 속성은 분석 및 모델 성능 향상 차원에서 제거하였다.

## 2025 전기 졸업과제 중간보고서

- 이후에도 모델 성능이나 분석 목적에 따라, 중요도 순위가 더 낮은 하위 속성들을 추가로 검토하여 점진적으로 제거해 나갈 예정이다.

### 5.3 데이터 증강 및 언더샘플링

#### 5.3.1 데이터 증강 – 윈도우 슬라이싱 + 노이즈

```
if (client_id, card_id) not in client_card_set:
    continue
for start in range(0, len(group)-WINDOW_SIZE+1, STRIDE):
    window = group.iloc[start:start+WINDOW_SIZE].copy()
    if window['fraud'].sum() > 0:
        valid_flag = True
        for _, row in window.iterrows():
            if (row['zip'], row['merchant_state'], row['merchant_city']) not in zip_combo_set or row['merchant_state'] != row['merchant_city']:
                valid_flag = False
                break
        if valid_flag:
            for i, row in window.iterrows():
                if row['fraud'] == 1:
                    new_row = row.copy()
                    # amount 노이즈
                    new_row['amount'] = add_amount_noise(row['amount'])
                    # date 노이즈
                    new_row['date'] = add_time_noise(row['date'])
                    # use_chip 변형 없이 그대로 유지 (온라인 거래는 반드시 Online Transaction, 아닌 경우 ch
                    augmented.append(new_row)
```

- 사기 거래 데이터의 불균형 문제를 완화하고 시계열적 맥락을 강화하기 위해 데이터 증강 기법을 적용하였다. 본 증강 과정에서는 윈도우 슬라이싱(window slicing) 기법과 속성별 노이즈 삽입 방식을 결합하여, 시계열 구조를 보존하면서도 데이터의 다양성을 높이하고자 하였다.
  - 이 과정에서 zip, state, city와 같이 속성 간 논리적 제약(예: 같은 zip에는 같은 state/city가 매칭됨)을 준수하였고 존재하지 않는 고객 ID나 카드 ID가 생성되지 않도록 사전에 데이터 정합성을 확보하였다.
  - 또한 거래 시간, 금액 등 일부 속성에는 적절한 범위 내에서 노이즈를 부여하여, 현실적 변동성과 데이터의 확장성을 동시에 반영하였다.
  - 이를 통해 데이터의 시계열적 특성을 보존하면서도 사기 거래 탐지 모델의 효율적 학습을 위한 샘플을 추가로 생성하였다.
- 먼저, 거래 데이터 내에서 고객-카드 단위로 연속적인 거래 시퀀스를 구축한 후, 슬라이딩 윈도우(예: 5개 거래, stride=1)를 적용하여 여러 개의 시계열 구간(윈도우)을 추출하였다.
- 각 윈도우에서 사기 거래가 포함된 경우, 해당 사기 거래를 복사하면서 거래 발생 시각(date)과 거래 금액(amount)에 대해  $\pm 10\sim 30$ 분,  $\pm 10\%$  이내의 무작위 노이즈를 각각 추가하여 현실적인 데이터 다양성 및 시계열 패턴 변화를 부여하였다.

## 2025 전기 졸업과제 중간보고서

- 이러한 방식은 실제로 존재하는 거래 흐름의 물리적·업무적 제약 조건(우편번호, 카드 소유, 상점 정보 등)은 100% 보존하면서, 새로운 시퀀스 맥락 및 사소한 수치 변화가 반영된 다양한 사기 거래 패턴을 생성하는 효과가 있다.
- 생성된 증강 사기 거래 데이터는 원본 사기 거래의 최소 4배 이상 확보될 때까지 반복하며, 충분한 데이터를 확보하도록 하였다.

### 5.3.2 데이터 언더샘플링 – KNN 기반 사기와 유사한 거래 우선 샘플링

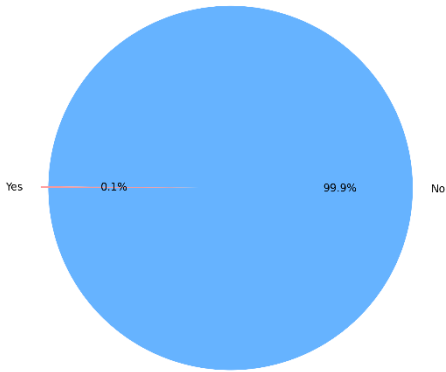
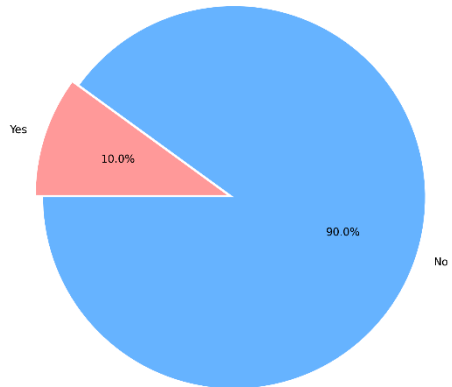
```
neighbors = NearestNeighbors(n_neighbors=1, metric='euclidean', n_jobs=-1)
neighbors.fit(X_normal)
distances, indices = neighbors.kneighbors(X_fraud)

indices_set = set(indices.flatten())
```

- 전체 거래 데이터에서 정상 거래가 대다수를 차지하므로, 사기 거래 탐지 모델이 사기 패턴에 효과적으로 집중하도록 정상 거래에 대한 언더샘플링 기법을 적용하였다.
- 단순 임의 추출 방식 대신, 증강된 사기 거래샘플들과 거래 금액, zip, mcc 등 주요 속성값이 유사한 정상 거래를 우선적으로 선택하기 위해 K-NN(최근접 이웃) 알고리즘을 활용하였다.
- 증강 사기 거래 각각에 대해 정상 거래 데이터에서 유클리드 거리 기준 가장 유사한 레코드를 탐색하여, 사기-정상 간 특성 분포가 최대한 비슷하도록 표본을 구성하였다.
- 선택된 정상 거래 표본이 목표치(사기 거래 데이터의 9배)에 미치지 못할 경우, 잔여 정상 거래에서 무작위 추출을 추가로 수행하여 최종 언더샘플링된 정상 거래 집합의 규모를 확보하였다.

## 2025 전기 졸업과제 중간보고서

### 5.3.3 증강 및 언더샘플링 전 · 후 결과

증강 및 언더샘플링 전 정상 및 사기 거래	증강 및 언더샘플링 후 정상 및 사기 거래												
<pre> 값이 있는 행 개수: 8914963 fraud No      8901631 Yes      13332 Name: count, dtype: int64 </pre>	<pre> 값이 있는 행 개수: 798690 fraud No      718821 Yes      79869 Name: count, dtype: int64 </pre>												
<p>Fraud (Yes/No) Distribution</p>  <table border="1"> <thead> <tr> <th>Category</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>No</td> <td>99.9%</td> </tr> <tr> <td>Yes</td> <td>0.1%</td> </tr> </tbody> </table>	Category	Percentage	No	99.9%	Yes	0.1%	<p>Fraud (Yes/No) Distribution</p>  <table border="1"> <thead> <tr> <th>Category</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>No</td> <td>90.0%</td> </tr> <tr> <td>Yes</td> <td>10.0%</td> </tr> </tbody> </table>	Category	Percentage	No	90.0%	Yes	10.0%
Category	Percentage												
No	99.9%												
Yes	0.1%												
Category	Percentage												
No	90.0%												
Yes	10.0%												

## 2025 전기 졸업과제 중간보고서

### 5.4 데이터 위치 정보 변환

(우편번호, 주(또는 해외국가), 도시 => 위도/경도 좌표)

```
if merchant_state in us_states_and_territories:
    # 미국 내 거래: zip 우선, 실패 시 city/state
    query = f"{zip_code}, USA"
    try:
        location = geocode(query)
    except:
        location = None

    if location is None:
        query = f"{merchant_city}, {merchant_state}, USA"
        try:
            location = geocode(query)
        except:
            location = None
else:
    # 해외 거래: city/state
    query = f"{merchant_city}, {merchant_state}"
    try:
        location = geocode(query)
    except:
        location = None
```

zip	merchant_state	merchant_city	latitude	longitude
10001	NY	New York	40.7484394	-73.9940079
10002	NY	New York	40.7170774	-73.9893192
10003	NY	New York	40.731341	-73.9887535
10004	NY	New York	40.7011717	-74.0135754
10005	NY	New York	40.7185829	-74.0069839
10006	NY	New York	40.708381	-74.013408
10007	NY	New York	40.7140653	-74.0083242
10008	NY	New York	40.7124137	-74.0104582
10009	NY	New York	40.7258747	-73.9808522
10010	NY	New York	40.7398623	-73.9851932
10011	NY	New York	40.7409101	-73.9995538

- 거래 데이터의 각 기록에 대해 거래 지점의 위도(latitude)와 경도(longitude) 좌표 정보를 부여하기 위해, 위치 정보 변환을 수행하였다.
- 미국 내 거래는 우선적으로 zip(우편번호)을 기반으로 위도·경도 좌표를 조회하였다.



## 2025 전기 졸업과제 중간보고서

- 만약 zip 코드로 좌표를 찾을 수 없거나 미국 외 해외 거래의 경우에는 merchant\_state(주 또는 국가)와 merchant\_city(도시) 값을 활용하여 해당 위치의 좌표를 검색하였다.
- 이 과정에는 지리좌표 변환 라이브러리(예: geopy)를 활용해서 zip, 주, 시티, 국가명 등 다양한 방식으로 정확도 높은 위치 조회가 이루어지도록 처리하였다.
- 조회된 위도·경도 값은 모든 거래 데이터에 추가하여 거래 간 위치 변화 및 이동 거리(예: 연속 거래 사이 공간적 거리, 급격한 위치 이동 탐지) 등 시계열·공간 분석 및 이상 거래 패턴 검출에 핵심 변수로 활용할 계획이다.
- 현재 변환된 위치 데이터는 csv 파일 형태로 저장되어 모델 학습 시 매핑되어 사용될 예정이다.
- geopy의 한계점: 거래 데이터 내 일부 장소(주소)의 경우 geopy 라이브러리를 활용한 자동 좌표(위도/경도) 변환이 실패하는 사례가 일부 발생하였다.

zip	merchant_state	merchant_city	latitude	longitude
00000	ONLINE	ONLINE	-1	-1
TLS00	East Timor (Timor-Leste)	Dili	-1	-1
ZAF00	South Africa	Johannesburg	-1	-1

- 이러한 경우에는 LatLong.net 등 외부 서비스에서 해당 정보를 수동으로 검색하여 좌표값을 보완하여 기입하였다.
- 향후 새로운 거래 데이터에서 geopy를 사용해도 좌표가 조회되지 않는 장소가 추가로 나타날 가능성이 존재한다. 이에 따라 보다 정밀하고 신뢰도 높은 좌표 변환이 가능한 대안 라이브러리(예: Google Maps API 등)의 도입을 검토하여 자동화 및 데이터 정확성 확보를 위한 추가 개선을 추진할 예정이다.
- 추후에는 백엔드 시스템과 연동하여 변환된 모든 위치 데이터를 DB에 저장하고, 새로운 거래 발생 시 실시간으로 위치 정보가 조회될 수 있도록 자동화 및 데이터 관리 시스템 확장도 계획 중이다.