[정보보안 HW4 - 실습보고서]

Anomaly Detection 과제

[정보컴퓨터공학부 201924437 김윤하]

---

1. OCSVM 실습 결과

[실습 완료 코랩 링크]

https://colab.research.google.com/drive/1Or2Q_c8d-aTTgGoapF_BaYO771OeO3Vn?usp=sharing

[추가 실습 화면]

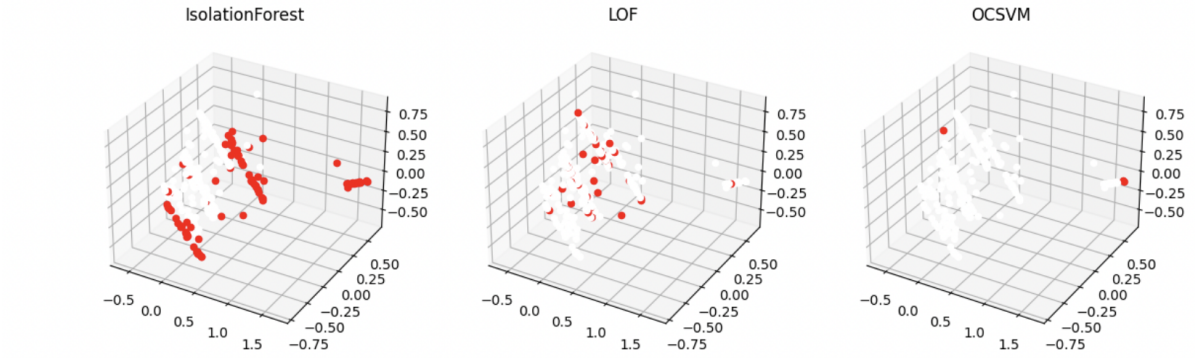
In [20]:
```
from sklearn import svm

clf = svm.OneClassSVM(nu = 0.01, kernel = 'rbf', gamma = 0.00001)
clf.fit(compressed_data)
pred_ocsvm = clf.predict(compressed_data)
cnt_ocsvm = collections.Counter(pred_ocsvm)
print(cnt_ocsvm)
```

```
Counter({1: 305319, -1: 2205})
```

In [23]:
```
import matplotlib.pyplot as plt # matplotlib을 이용해 결과가 잘 보이도록 3D 모델링

fig = plt.figure(figsize=(14, 6))
res_if = fig.add_subplot(131, projection='3d')
res_if.set_title('IsolationForest')
res_lof = fig.add_subplot(132, projection='3d')
res_lof.set_title('LOF')
res_ocsvm = fig.add_subplot(133, projection='3d')
res_ocsvm.set_title('OCSVM')
for i in range(0, len(compressed_data), 1000):
  res_if.scatter(compressed_data[i][0], compressed_data[i][1], compressed_data[i][2], c='white' if pred_if[i] ==
  res_lof.scatter(compressed_data[i][0], compressed_data[i][1], compressed_data[i][2], c='white' if pred_lof[i] =
  res_ocsvm.scatter(compressed_data[i][0], compressed_data[i][1], compressed_data[i][2], c='white' if pred_ocsvm[
plt.show()
```



2. 실습 결과 분석

먼저, **OCSVM**은 **One-Class Support Vector Machine**으로 하나의 클래스를 기반으로 작동하는 솔루션입니다. 이 알고리즘은 주어진 데이터의 대부분이 정상 클래스에 속하고 드물게 발생하는 이상 클래스를 감지하는 것을 목표로 합니다. OCSVM은 원점으로부터 데이터를 분리하는 초평면(hyperplane)을 찾는 방식으로 작동합니다.

실습에서 사용한 **Isolation Forest**와 **Local Outlier Factor**의 결과에서는 각각 **75563개**와 **34917개**의 이상치로 분류된 데이터가 나왔지만, OCSVM에서는 **2205개**의 이상치만 찾아냈습니다. OCSVM이 다른 모델에 비해 적은 이상치를 탐지한 이유에는 여러 가지 이유가 있을 수 있습니다.

OCSVM은 정상 데이터를 찾는 것을 주 목적으로 하기 때문에 이상치를 상대적으로 적게 탐지하는 경향이 있습니다. OCSVM은 정상 데이터의 영역을 찾고, 이상치는 해당 영역 밖에 위치하게 됩니다. 따라서 이상치를 정확하게 탐지하기보다는 정상 데이터를 더 중요시하는 경향이 있습니다.

또한, 데이터의 차원과 분포에 따라 OCSVM의 성능과 결과가 달라질 수 있습니다. 데이터의 차원이 높거나 분포가 복잡한 경우 OCSVM의 실행 시간이 오래 걸릴 수 있습니다. 또한, OCSVM은 γ (gamma) 매개변수 값을 조정하여 실행 속도와 성능 사이의 균형을 조절해야 합니다.

반면에 Isolation Forest는 데이터의 밀도 기반으로 이상치를 탐지하는데, 트리 기반의 분할을 통해 이상치를 식별합니다. 따라서 속도가 빠르고 많은 이상치를 탐지할 수 있습니다. 또한, Local Outlier Factor는 데이터의 군집과 밀도를 고려하여 이상치를 판단하는데, 이로 인해 극소 범위까지 고려할 수 있습니다.

따라서 OCSVM은 다른 알고리즘에 비해 이상치를 적게 탐지하는 경향이 있으며, 이는 OCSVM의 작동 방식과 데이터의 특성에 기인합니다. 적절한 이상치 탐지 알고리즘을 선택하기 위해서는 데이터셋의 특성과 목적에 맞는 알고리즘을 선택하는 것이 중요합니다.