[1] Differentiation of the Proposed Model

Table 1 summarizes the features of the proposed model in terms of its application to FDS, compared to existing anomaly detection methods. In <Table 3-1>, supervised and unsupervised refer to supervised-based and unsupervised-based anomaly detection methods, respectively. Few Shot (1) represents the first Few Shot Anomaly Detection method explained in Sections 2-3, which is the Anomaly-Dependent Feature Learning approach. Few Shot (2) refers to the DevNet method explained second in Section 2-3. Finally, GenFSAD represents the proposed model in this study.

**Table 1.** Table comparing the distinguishing features of the proposed models.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Supervised Learning | Unsupervised Learning | Few Shot(1) | Few Shot(2) | GenFSAD |
| Need for data labeling | Required | Not  required | Requires anomaly labels | Requires anomaly labels | Requires anomaly labels |
| Suitability for tabular data | Applicable | Not  applicable | Not  applicable | Applicable | Applicable |
| Use of anomaly data | Used | Not Used | Used | Used | Used |
| Incorporation of degree of normality | Not incorporated | Not incorporated | Partially incorporated | Not incorporated | Incorporated |
| Distribution assumption | No  assumption | No  assumption | No  assumption | Assumption | No  assumption |

* Cost only for labeling a small amount of data: The existing supervised anomaly detection methods had the problem of high data labeling costs because both normal and anomalous data had to be labeled. GenFSAD also requires data labeling for a small number of anomalies, making it more costly than unsupervised anomaly detection. However, for "normal data or unlabeled data," it is not necessary to label the data because the Genuine Detector re-labels them.
* Suitable for tabular data: Unsupervised anomaly detection methods and anomaly-dependent feature learning through Two Step were not good at handling tabular data due to the architecture's structure. However, GenFSAD is an end-to-end anomaly detection learning methodology that directly learns anomaly scores, making it suitable for FDS that uses tabular data.
* Use of outlier data: The non-use of outlier data in unsupervised learning resulted in an increase in false positive rates. However, GenFSAD is a Few Shot Anomaly Detection model that utilizes outliers with prior/partial information to learn data abnormality. ·
* Consideration of Genuineness: GenFSAD is a model that takes into account the possibility of missing outlier data in the FDS process and relabels training data based on the degree of genuine. · No distribution assumption: DevNet assumed a distribution, which was a limitation when using small datasets, but GenFSAD is a methodology that can be used on any data without making assumptions about its distribution.

Experimental setup and Results

[2] Data Description

In this study, six datasets were utilized to verify whether the proposed model GenFSAD is a suitable methodology for outlier detection, including FDS. Four datasets, credit card, synthetic, car claims, and loan default are related to abnormal transactions in the financial field, while the other two datasets, marketing and thyroid, are commonly used in general anomaly detection research.

**Table 2.** Dataset Description Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data | Number  of data | Dimension | Ratio(%) | Objective | Format | Field |
| credit card | 284,807 | 31 | 0.01 | Credit card fraud detection | Tabular | Financial |
| synthetic | 6,362,620 | 11 | 0.12 | Mobile fraud detection | Tabular | Financial |
| car claims | 15,420 | 33 | 4.01 | Automobile insurance fraud detection | Tabular | Financial |
| loan default | 148,670 | 33 | 24.60 | Loan default prediction | Tabular | Financial |
| marketing | 41,888 | 21 | 0.11 | Term deposit subscription prediction | Tabular | Non-  financial |
| thyroid | 7,200 | 21 | 0.55 | Thyroid cancer diagnosis | Tabular | Non-  financial |

### [3] Introduction of comparison models

This study assesses the proposed model's performance by comparing it with GenFSAD and five representative models in each anomaly detection field.

* DevNet: DevNet (Deviation Network) is an End-to-End Anomaly Scoring model that directly produces Anomaly Score using Gaussian prior distribution and currently shows state-of-the-art performance in the FDS detection field, which is the target of this study (Pang, Shen, and Hengel. 353-362).
* DeepSAD: Deep SAD (Deep Semi Supervised Anomaly Detection) is a feature learning methodology based on Anomaly Measure Dependent Feature Learning that utilizes a small number of anomalies and is currently a state-of-the-art model in the One-Class Classification-based anomaly detection field (Ruff et al. "Deep Semi-supervised" 1-12).
* REPEN: REPEN is a distance-based triplet neural network anomaly detection methodology that utilizes a small number of anomalies. It has a common feature with the proposed model in that it uses the Random Nearest Neighbor Distance-based Detector (Pang et al. "Learning" 2041-2050).
* AutoEncoder: AutoEncoder-based anomaly detection is a research methodology that has shown good performance in the field of deep learning-based unsupervised anomaly detection. It has a feature of deriving anomaly scores using Reconstruction Error (Chen et al. 90-98).
* iForest: iForest(Isolation Forest) is a widely used unsupervised anomaly detection method that assigns anomaly scores based on the number of branches of decision trees used to find the values of the data points (Liu et al. 413-422).

### In the experiments, all models were implemented in Python. GenFSAD used the Keras library, while DevNet and REPEN were also implemented with the Keras library according to the authors' descriptions. The DeepSAD model, though using the author's code, was implemented in the PyTorch library. AutoEncoder and iForest were implemented using the scikit-learn package.

* **[4] Experiment Parameter Setting.**
* In this study, GenFSAD utilized a sub-sample size of 8 and an ensemble iteration of 50 for Data Re-Labeling (Sp parameter), following the implementation from the referenced paper (Sugiyama and Borgwardt 5-6). The outlier score for each data instance was calculated using the k-d tree and Euclidean distance function from the scikit-learn package. In the training process, GenFSAD, DevNet, REPEN, and DeepSAD, which are Few-Shot outlier detection models using neural network structures, were trained with a single hidden layer of 20 units in a multi-layer perceptron network (MLP) structure (Pang, Shen, and Hengel 357; Pang et al. "Deep Weakly-supervised" 13). ReLU was used as the activation function, and a common l2-norm regularizer of λ=0.01 was applied to prevent overfitting. Gradient descent used the Root Mean Square Propagation (RMSprop) optimizer (Tieleman and Hinton 26-31), with a batch size of 1024 and 20 training epochs. For AE, normal feature learning employed two MLP hidden layers each for the encoder and decoder. ReLU served as the activation function, and an l2-norm regularizer of λ=0.01 prevented overfitting. Gradient descent used the Adam optimizer, with a batch size of 32 and 20 training epochs. The iForest model, which does not use a neural network structure, followed scikit-learn package settings, training with a sub-sample size of 256 and 100 iterations in the ensemble process.

[5] Experimental Results

### Performance Comparison of Each Model

**Table 3.** AUC-ROC Performance Table for Each Model (mean ± standard deviation)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AUC-ROC Performance | | | | | | |
| Data | GenFSAD | DevNet | DeepSAD | REPEN | AE | IF |
| credit card | 0.976±0.005 | 0.963±0.004 | 0.946±0.002 | 0.945±0.003 | 0.954±0.001 | 0.900±0.003 |
| synthetic | 0.949±0.009 | 0.929±0.017 | 0.900±0.004 | 0.872±0.024 | 0.830±0.023 | 0.742±0.021 |
| car claims | 0.820±0.007 | 0.789±0.004 | 0.634±0.004 | 0.508±0.078 | 0.558±0.011 | 0.502±0.004 |
| loan default | 1.000±0.000 | 1.000±0.000 | 1.000±0.000 | 0.799±0.005 | 0.936±0.003 | 0.594±0.017 |
| marketing | 0.938±0.002 | 0.903±0.010 | 0.784±0.019 | 0.723±0.006 | 0.698±0.008 | 0.625±0.009 |
| thyroid | 0.922±0.010 | 0.856±0.004 | 0.809±0.009 | 0.481±0.009 | 0.553±0.013 | 0.569±0.017 |
| Average | 0.934±0.005 | 0.907±0.006 | 0.846±0.006 | 0.722±0.021 | 0.755±0.010 | 0.655±0.012 |
| p-value |  | 0.043 | 0.043 | 0.028 | 0.028 | 0.028 |

**Table 4.** AUC-PR Performance Table for Each Model (mean ± standard deviation).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| AUC-PR Performance | | | | | | |
| Data | GenFSAD | DevNet | DeepSAD | REPEN | AE | IF |
| credit card | 0.515±0.049 | 0.681±0.110 | 0.563±0.046 | 0.604±0.005 | 0.201±0.004 | 0.020±0.001 |
| synthetic | 0.134±0.056 | 0.099±0.043 | 0.075±0.003 | 0.084±0.000 | 0.011±0.004 | 0.003±0.000 |
| car claims | 0.191±0.004 | 0.144±0.003 | 0.092±0.002 | 0.064±0.004 | 0.076±0.008 | 0.060±0.000 |
| loan default | 1.000±0.000 | 0.999±0.000 | 0.999±0.001 | 0.478±0.017 | 0.816±0.012 | 0.307±0.013 |
| marketing | 0.603±0.002 | 0.538±0.001 | 0.349±0.023 | 0.330±0.009 | 0.285±0.010 | 0.189±0.009 |
| thyroid | 0.661±0.016 | 0.494±0.007 | 0.398±0.012 | 0.074±0.026 | 0.088±0.007 | 0.112±0.016 |
| Average | 0.535±0.032 | 0.492±0.027 | 0.413±0.015 | 0.272±0.010 | 0.246±0.007 | 0.115±0.006 |
| p-value |  | 0.249 | 0.028 | 0.046 | 0.028 | 0.028 |