**DL**

**Autoencoder + CatBoost**

Датасет BAF

Linear (input size)

Linear (3904)

Linear (512)

Linear (64)

Linear (16)

Linear (512)

Linear (64)

Linear (16)

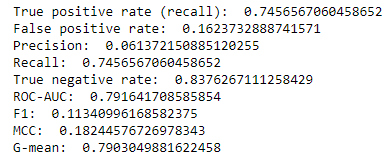
Linear (3904)

Linear (input size)

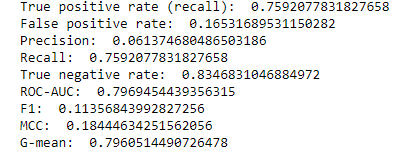
**CatBoost**

Рис. Архитектура Автоэнкодера

**Результаты**:



**Предыдущие результаты** **CatBoost** без автоэнкодера



Дальнейшие эксперименты проводились на датасете CCF

**MLP**

Linear (input size)

Linear (13)

Linear (input size)

Рис. Архитектура MLP

Loss – nn.BCEWithLogitsLoss()

Optimizer – Adam(model.parameters(), lr=lr, weight\_decay=1e-4)

Learning rate – 5e-3

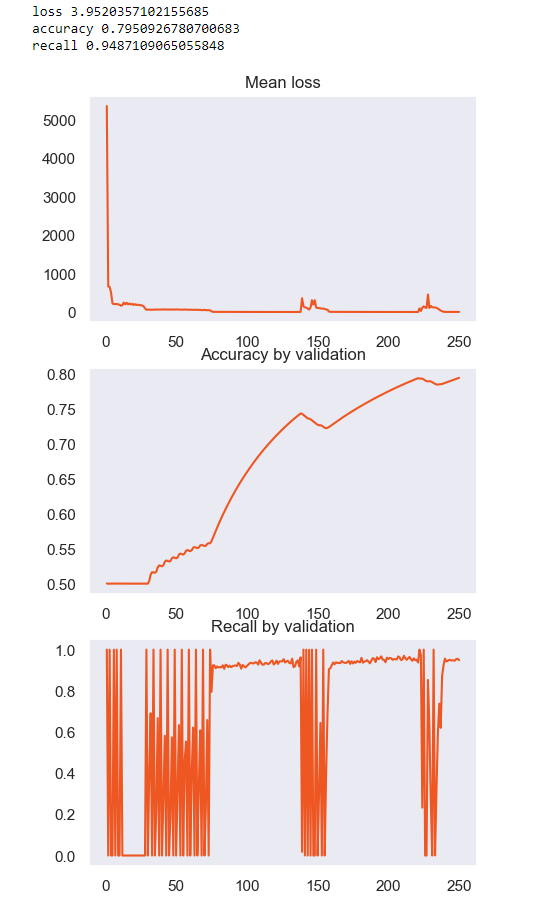
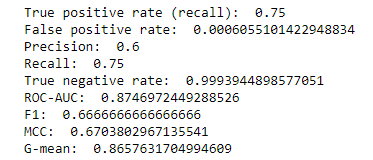
Epochs – 250

Рис. Обучение MLP

**Результаты**:

**CNN**

Loss – nn.BCEWithLogitsLoss()

Optimizer – SGD(model.parameters(), lr=lr, momentum=0.9)

Learning rate – 0.1

Sheduler – CyclicLR(optimizer, base\_lr=0.001, max\_ lr=0.1, step\_size\_up=3, mode="exp\_range", gamma=0.85)

Epochs – 400

Linear (input size)

Linear (128)

Conv1d (16, 32, kern=2)

Linear (1024)

Linear (4096)

Reshape

Conv1d (32, 64, kern=2)

Conv1d (64, 64, kern=1)

Conv1d (64, 64, kern=1)

Conv1d (64, 64, kern=1)

Conv1d (64, 64, kern=1)

Linear (16256)

Linear (128)

Linear (64)

Linear (output size)

Рис. Архитектура CNN

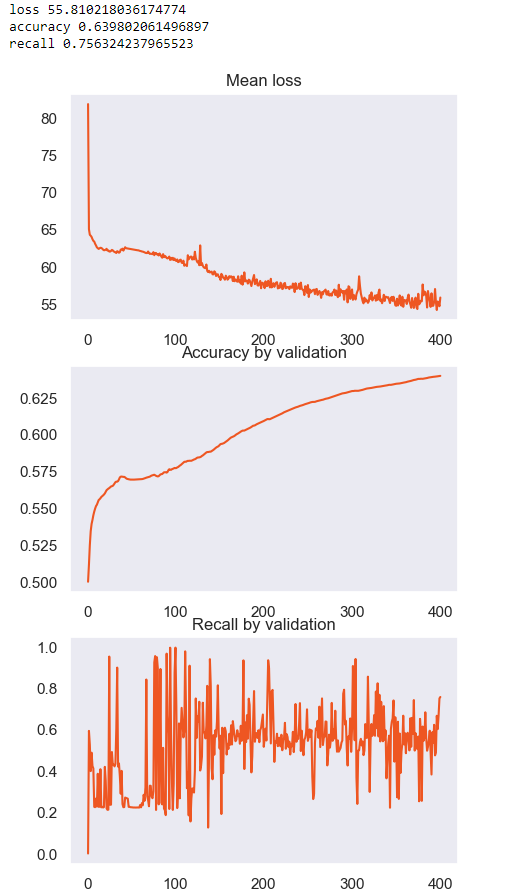
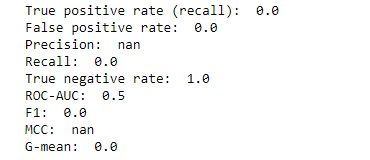


Рис. Обучение CNN

**Результаты**

**Экспериментальная модель**

Loss – nn.CrossEntropyLoss()

Optimizer – Adam(model.parameters(), lr=lr, weight\_decay=1e-4)

Learning rate – 0.1

Sheduler – OneCycleLR(optimizer, max\_lr=0.1, steps\_per\_epoch=2, epochs=epochs)

Epochs – 400

Linear (input size)

Linear (512)

Linear (512)

Linear (64)

Linear (16)

X

Linear (input size)

Concatenate

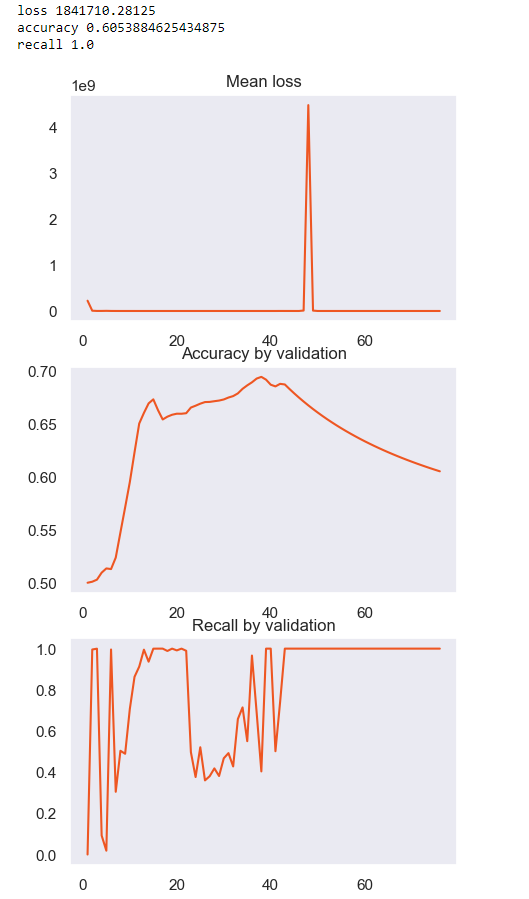
Linear (13)

Linear (output size)

Linear (16)

Linear (64)

Рис. Архитектура экспериментальной модели

Рис. Обучение экспериментальной модели

**Экспериментальная модель**

**с предварительно обученным автоэнкодером**

**Автоэнкодер**

Loss – nn.MSELoss()

Optimizer – Adam(model.parameters(), lr=lr, weight\_decay=1e-4)

Learning rate – 0.005

Sheduler – OneCycleLR(optimizer, max\_lr=0.1, steps\_per\_epoch=5, epochs=epochs)

Epochs – 20

**Классификатор**

Loss – nn.BCEWithLogitsLoss()

Optimizer – SGD(model.parameters(), lr=lr, momentum=0.9)

Learning rate – 0.001

Sheduler – CyclicLR(optimizer, base\_lr=0.001, max\_lr=0.1, step\_size\_up=5, mode="exp\_range", gamma=0.85, cycle\_momentum=True)

Epochs – 400



Рис. Обучение классификатора в экспериментальной модели

с предобученным автоэнкодером

**GAN**

**WGAN для датасета CCF**

Для нормализации данных использован QuantileTransformer

Балансировка данных – SMOTE

Linear (input size)

Linear (1000)

Linear (500)

Linear (output size)

Рис. Архитектура Генератора

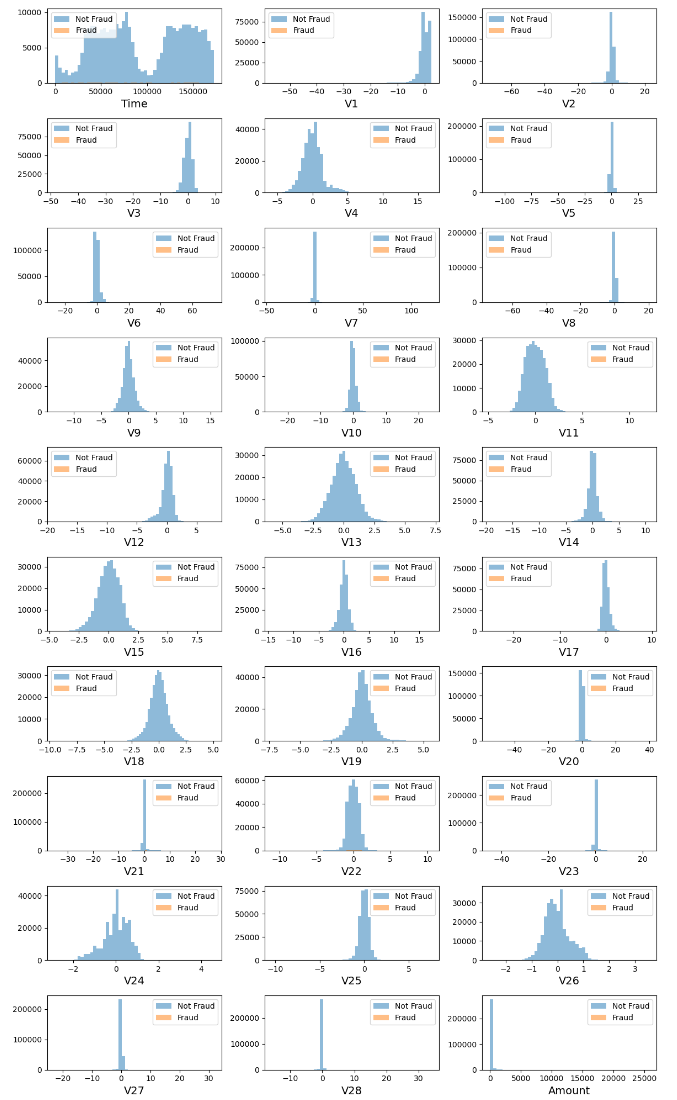
****

Рис. Распределения переменных в данных

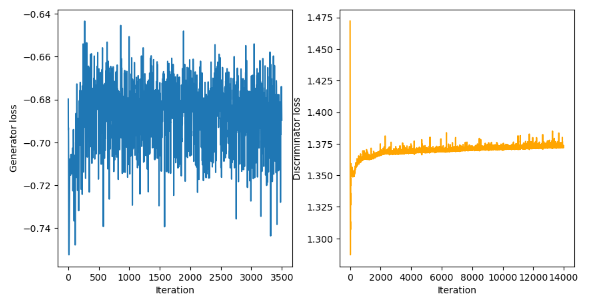
****

Рис. Обучение WGAN

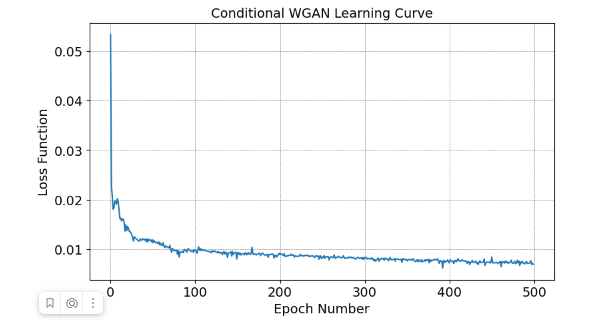
****

Рис. Обучение WGAN

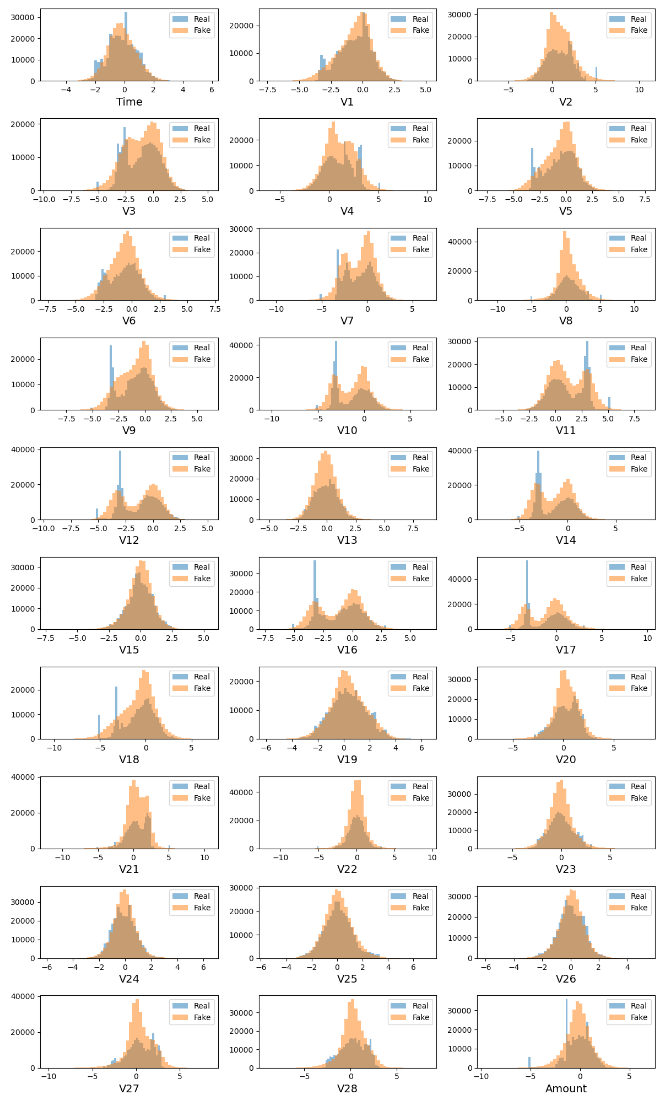


Рис. Результаты генерации нормализованные (оранжевым цветом)

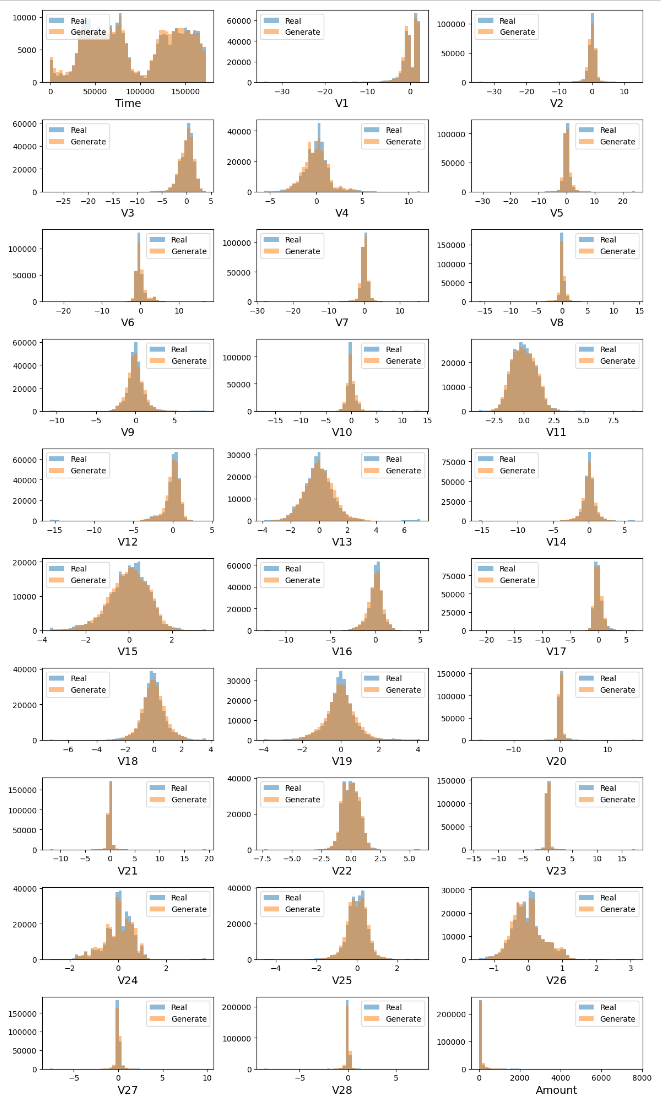


Рис. Результаты генерации исходные (оранжевым цветом)

ROC-AUC GradientBoostingClassifier (Fake, Real) – 0.9062