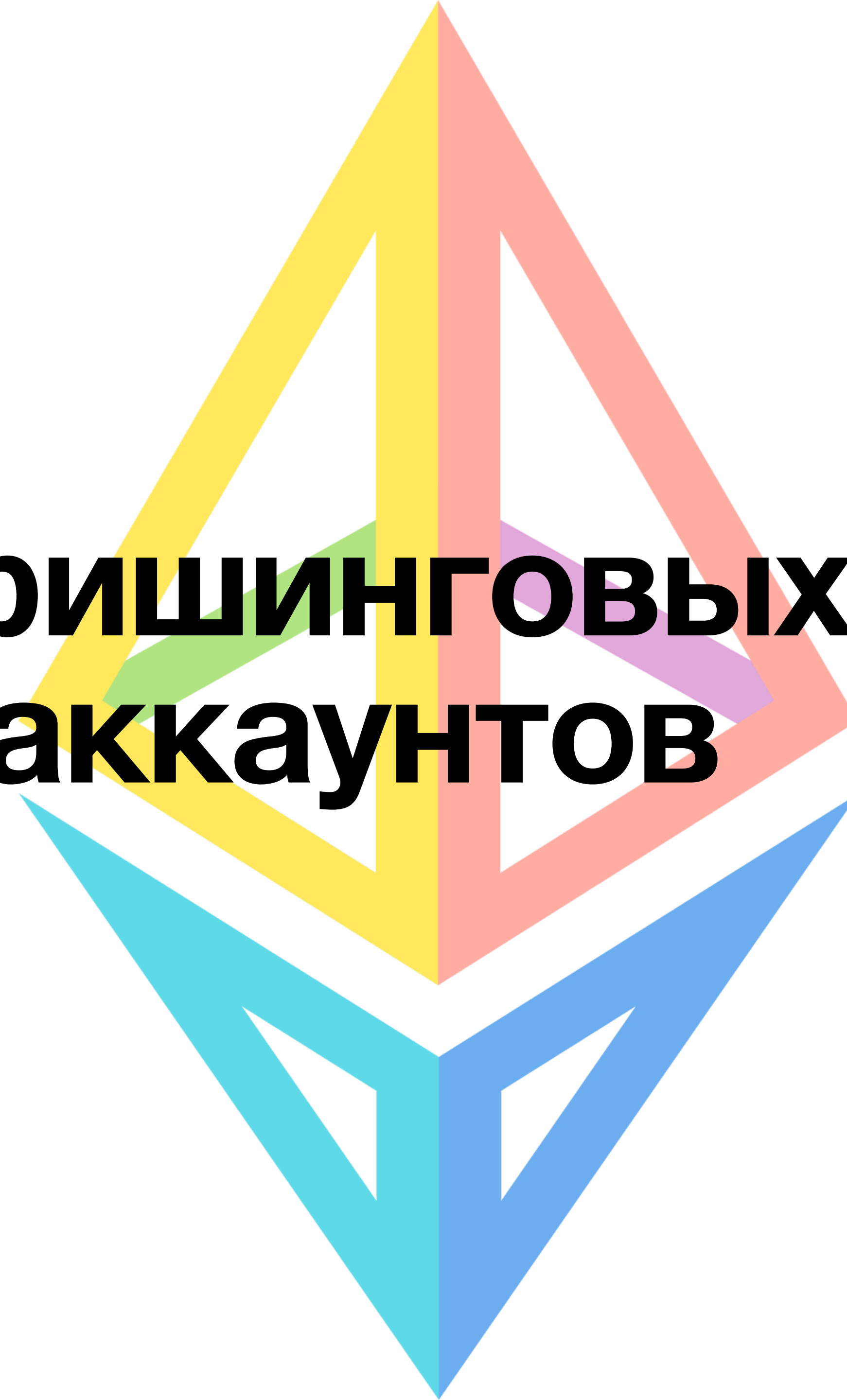


**Курсовая работа по теме**

# **Классификация фишинговых криптовалютных аккаунтов**

**Выродов Артём, ПИ20-3**



# Предметная область

- Блокчейн - технология децентрализованного обмена и хранения информации

- Криптовалюты - одна из реализаций блокчейна

- Ethereum - блокчейн платформа для создания приложений, имеющая собственную криптовалюту и систему аккаунтов.

The Ethereum logo is a 3D, faceted geometric shape, resembling a stylized 'E' or a crystal. It is composed of several triangular and quadrilateral faces in various shades of blue and white, giving it a three-dimensional appearance. It is centered on the slide.

ethereum

# Почему важно классифицировать фишинговые аккаунты

Фишинговые адреса

Создание поддельных транзакций и  
обман пользователей

Проведение атак на смарт-  
контракты

Создание мошеннических ICO и обман  
инвесторов

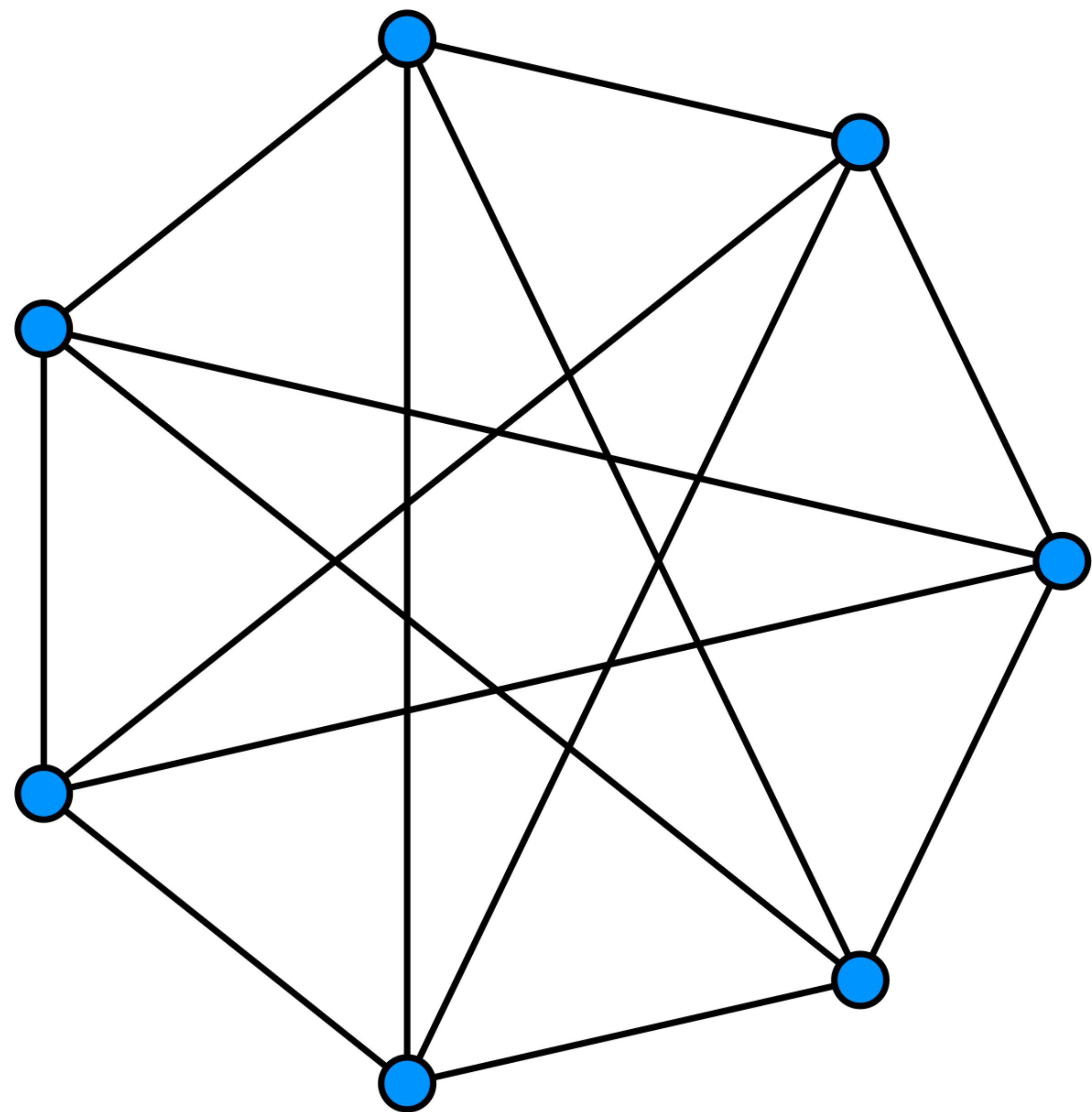
Классификация

Недопущение потери  
финансовых активов

Защита  
пользователей

# Набор данных

Представление	Граф, объект NetworkX, формат pickle
Тип графа	Направленный
Количество вершин (аккаунтов)	2973489
Количество рёбер (транзакций)	13551303
Средняя степень вершины (транзакции на аккаунт)	4.56
Атрибуты вершин	Да: фишинговый или нормальный
Атрибуты рёбер	Да: время и сумма транзакции





# Инструменты EDA



*matplotlib*

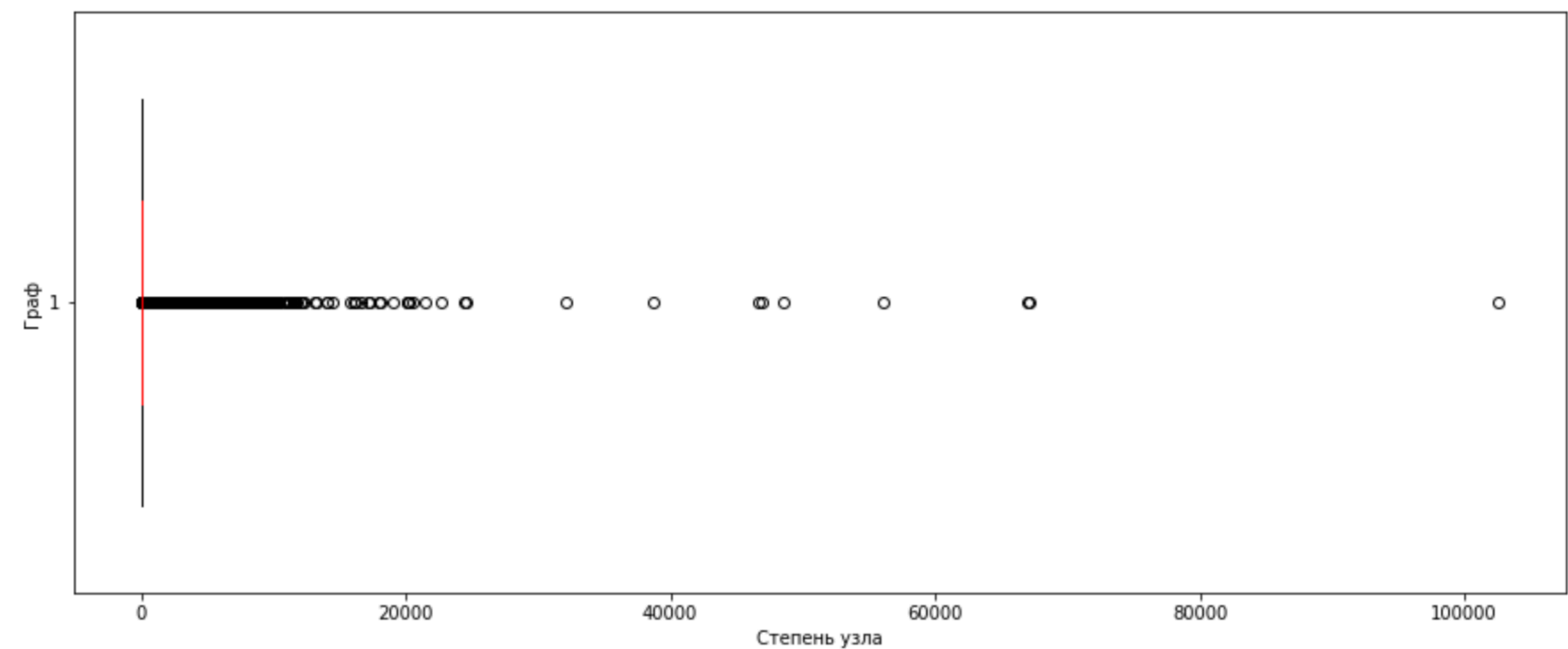
*Gp* Gephi

# 1165

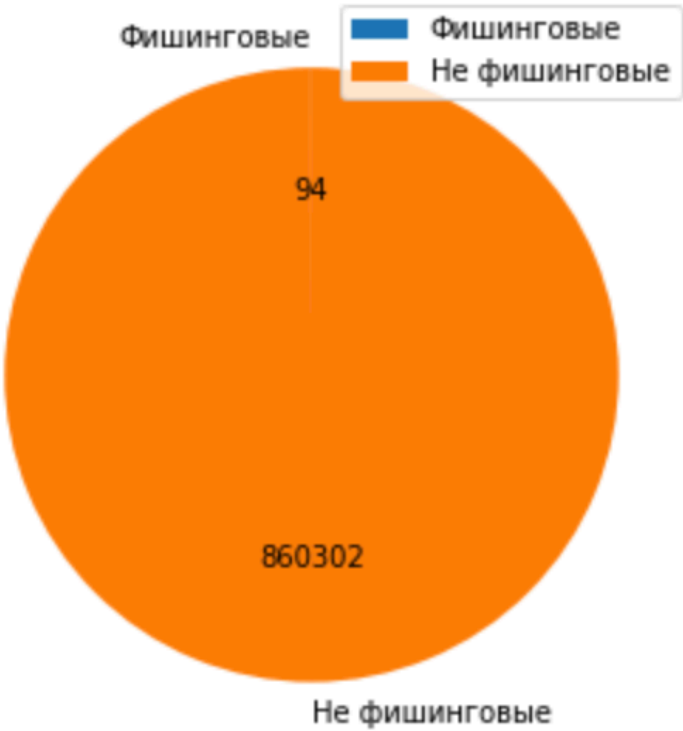
**фишинговых аккаунтов в датасете при общей численности 3 млн**

# Изучение признаков аккаунтов для уменьшения несбалансированности

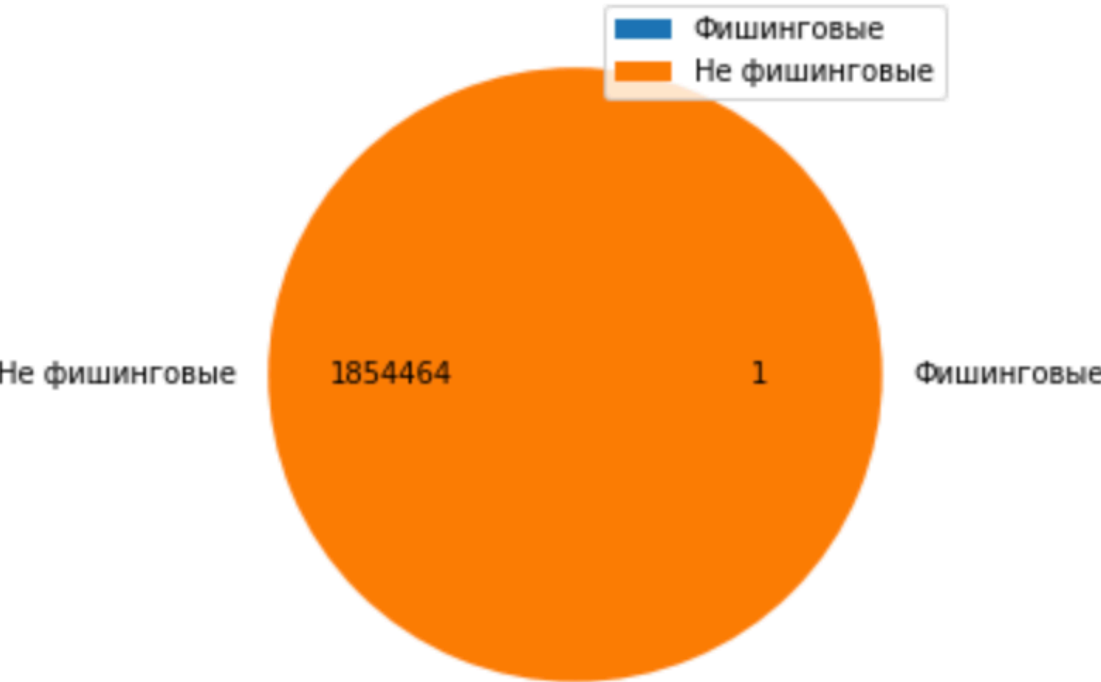
- 1. Анализ степеней вершин - количества проводимых транзакций
- 2. Поиск изолированных / без входящих / исходящих транзакций аккаунтов
- 3. Определение наличия циклов в графе
- 4. Определение наличия мультирёбер и петель
- 5. Выделение сообществ (подграфов)



Фишинговые аккаунты, не имеющие исходящих транзакций



Фишинговые аккаунты, не имеющие исходящих транзакций



# Факторы, выделяющие фишинговые аккаунты

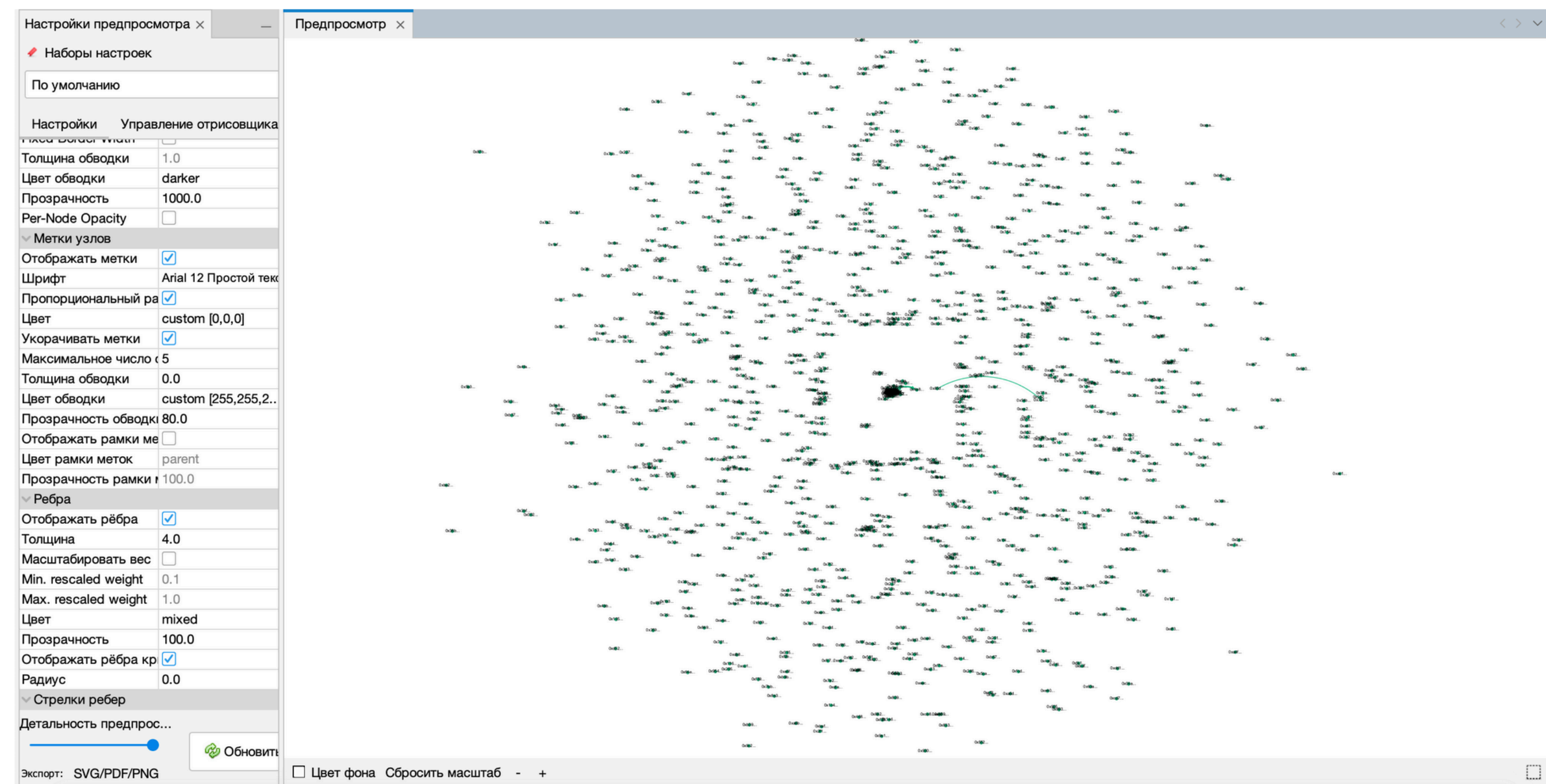
Признак изолированности	✗
Участие в повторяющихся транзакциях	✗
Переводы самому себе	✗
Средняя активность в сети	✓
Разбиение на подграф	✓





# Выделение сообществ алгоритмом Лувена

- Найдено 15574 сообщества, из которых 14841 - без фишинговых узлов
- После удаления подграфов без фишинговых узлов в графе осталось 553879 узлов и 2142657 рёбер
- Было удалено 81% узлов и 84% рёбер



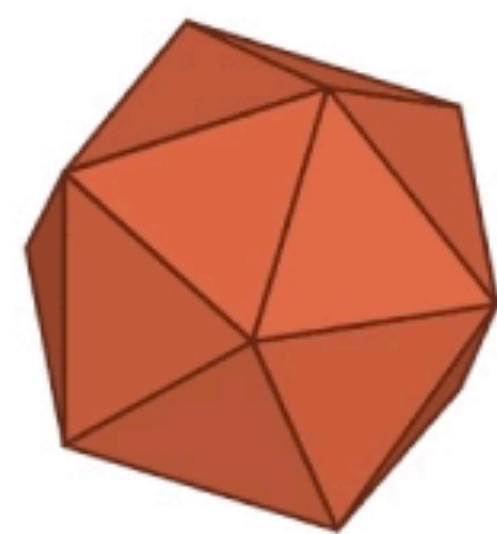
# Архитектуры нейронных сетей для обучения на графах

1. Графовые нейронные сети (Graph Neural Networks, GNNs) - класс моделей, разработанных специально для работы с графами. Они обрабатывают графы, используя сверточные слои, агрегирующие слои и пространственную инвариантность.
2. Модели на основе узловых эмбеддингов - модели, которые сначала преобразуют каждый узел в вектор, называемый узловым эмбеддингом, а затем используют эти эмбеддинги для предсказания атрибутов узлов. Примерами: node2vec и DeepWalk.
3. Модели на основе метрик - модели, которые используют метрики для измерения сходства между узлами в графе. Затем они используют эти метрики для предсказания атрибутов узлов. Примеры: модель на основе косинусного расстояния.

# GNNs



Vs

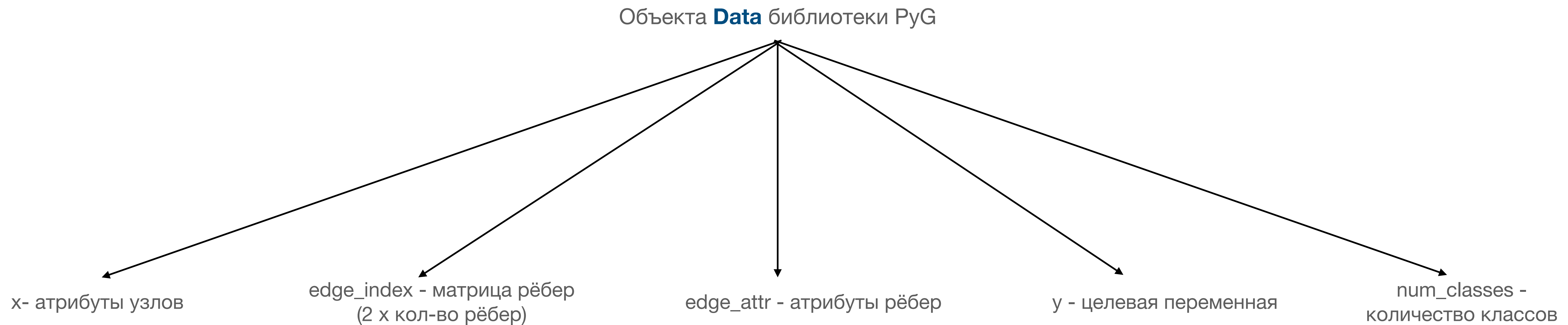


PyTorch  
geometric

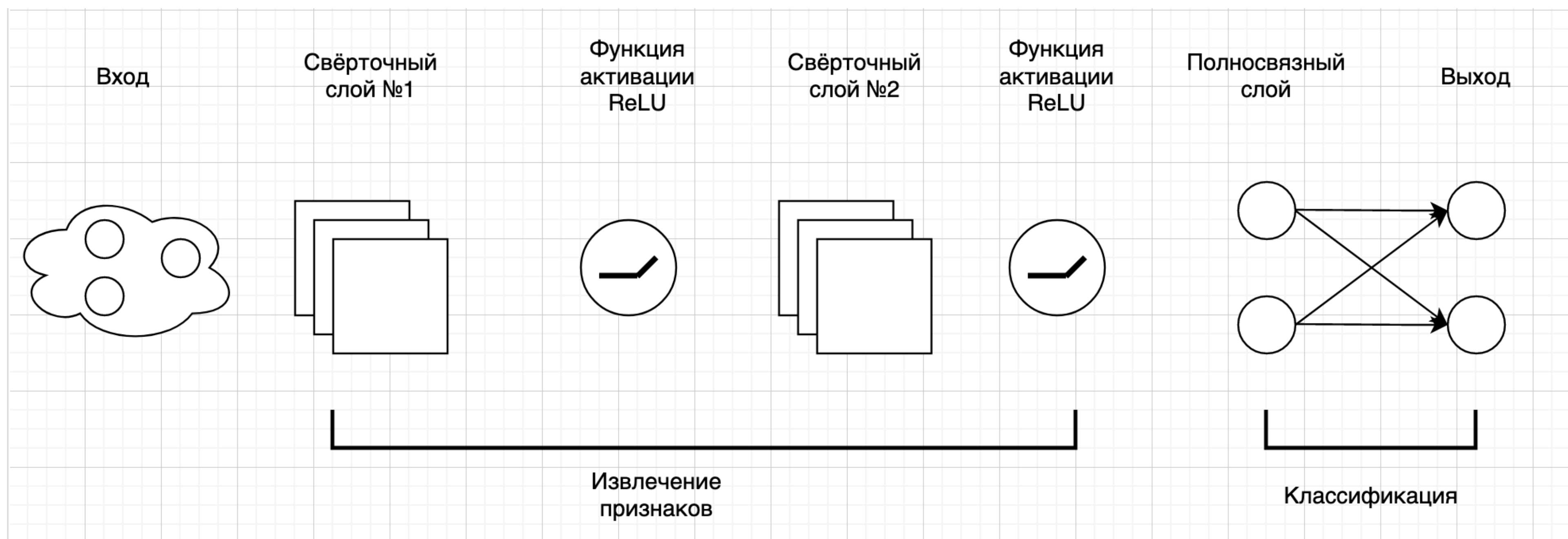
**Библиотеки для реализации GNN**



# Создание датасета PyG из графа NetworkX



# Модель №1. Графовая свёрточная нейронная сеть



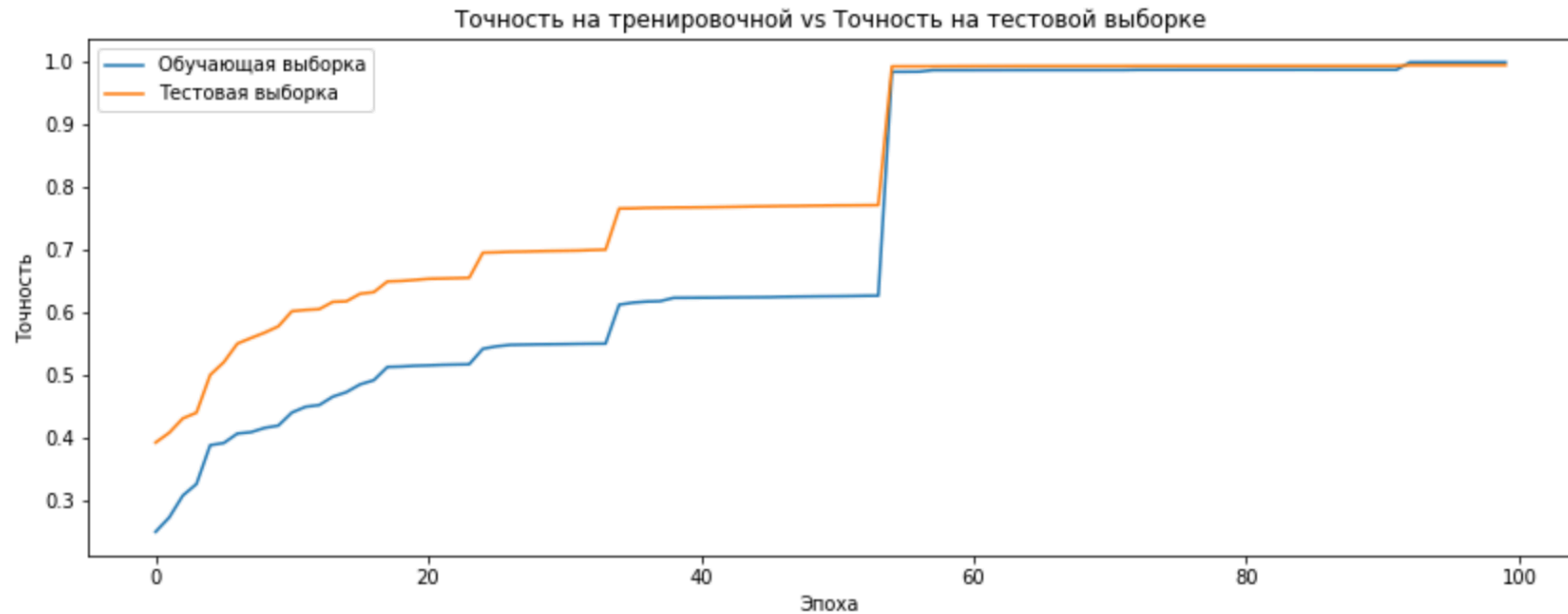
```
class GCN(torch.nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super(GCN, self).__init__()
        self.conv1 = GCNConv(input_dim, hidden_dim)
        self.conv2 = GCNConv(hidden_dim, output_dim)
        self.linear = Linear(output_dim, output_dim)

    def forward(self, data):
        x, edge_index, edge_weight = data.x.float(), data.edge_index, data.edge_attr[:,0]

        x = self.conv1(x, edge_index, edge_weight)
        x = torch.relu(x).float()
        edge_weight = data.edge_attr[:,1].float()
        x = self.conv2(x, edge_index, edge_weight)
        x = torch.relu(x)
        x = self.linear(x)
        return x
```

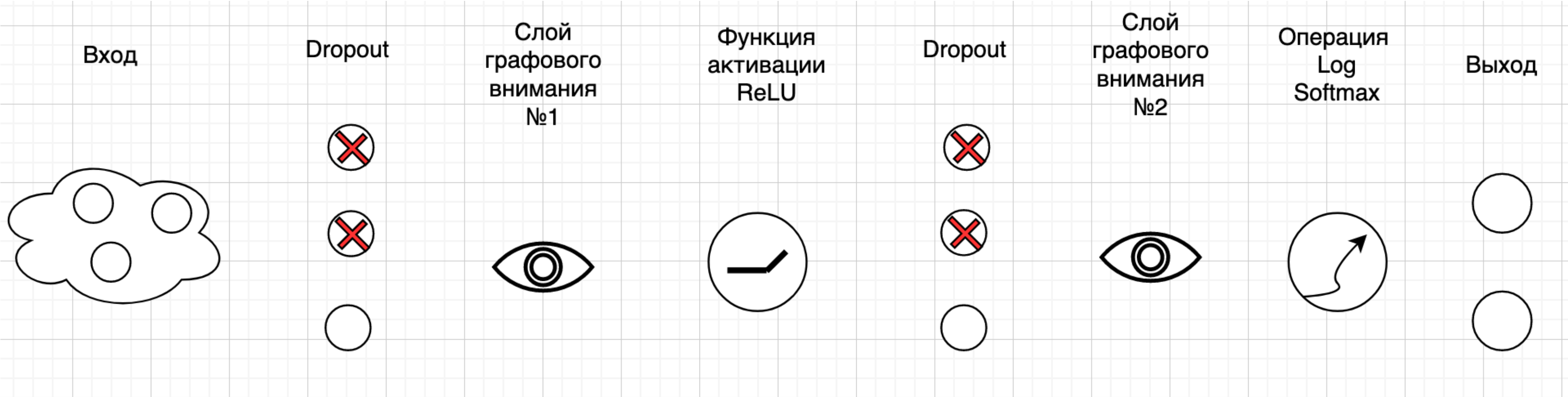


# Результаты GCN



Время обучения: 10 минут, Precision: 100%, Recall: 98%, F1-score: 99%

# Модель №2. Графовая сеть внимания



```
class GAT(torch.nn.Module):
    def __init__(self, data, heads_layer1, heads_layer2, dropout):
        super().__init__()
        self.dropout = dropout
        num_features = data.num_features
        num_classes = len(data.y.unique())

        self.conv1 = GATConv(in_channels=num_features, out_channels=8,
                              heads=heads_layer1, concat=True, negative_slope=0.2,
                              dropout=dropout)

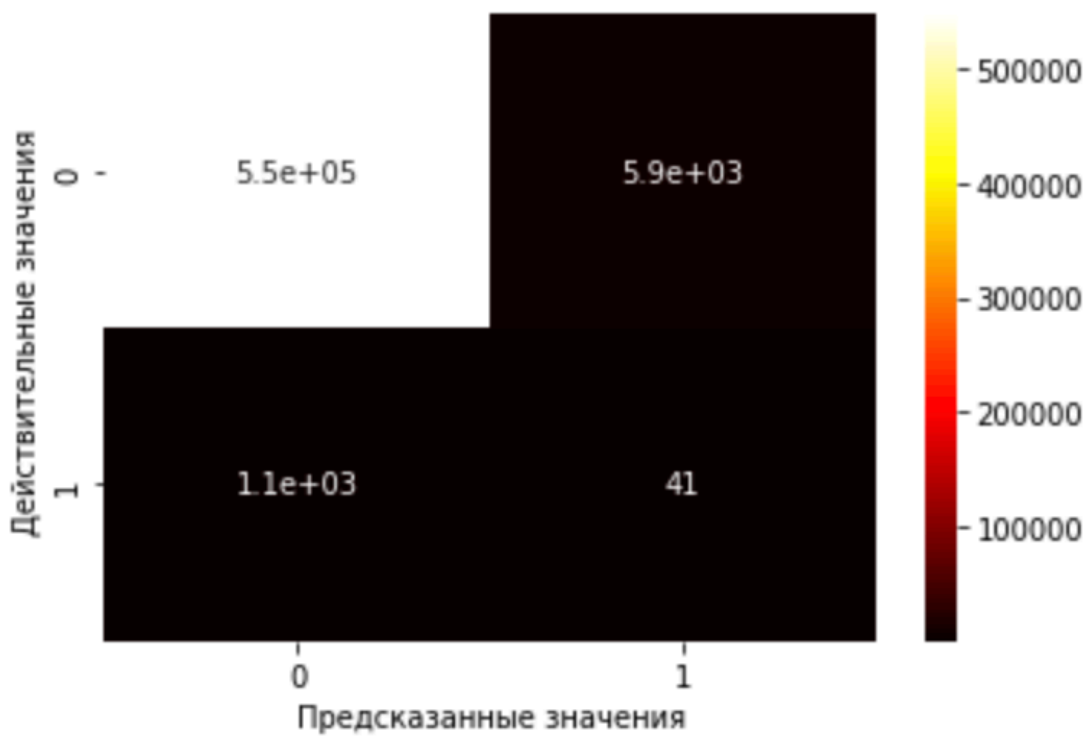
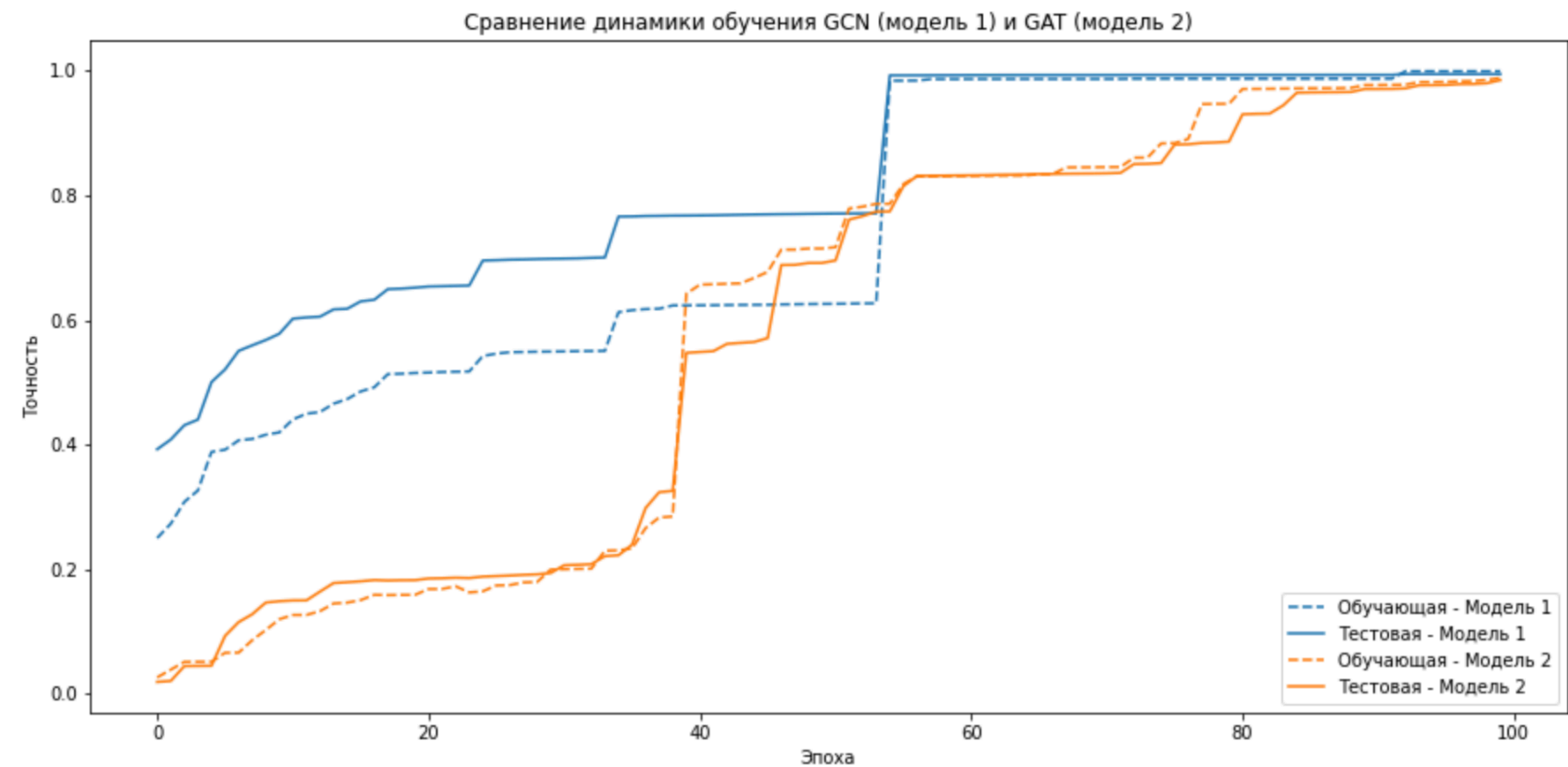
        self.conv2 = GATConv(in_channels=8*heads_layer1, out_channels=num_classes,
                              heads=heads_layer2, concat=False, negative_slope=0.2,
                              dropout=dropout)

    def forward(self, data):
        x, edge_index, edge_attr = data.x.float(), data.edge_index, data.edge_attr

        x = F.dropout(x, p=self.dropout, training=self.training)
        x = self.conv1(x, edge_index, edge_attr)
        x = F.elu(x)
        x = F.dropout(x, p=self.dropout, training=self.training)
        x = self.conv2(x, edge_index, edge_attr)

        return F.log_softmax(x, dim=1)
```

# Результаты GAT



Верно идентифицировано  
фишинговых аккаунтов:  
**3.5%**

Время обучения: 10 минут, Precision: 100%, Recall: 96%, F1-score: 98%

# Итоги

1. Данные, хранимые в виде графа, удобно обрабатывать и исследовать с помощью библиотеки NetworkX
2. Транзакционный граф Ethereum, в виду своего масштаба представляет из себя крайне несбалансированный набор данных, соотношение классов в котором должно быть выравнено путём удаления большого числа экземпляров данных (которые выбираются классическими графовыми алгоритмами, например такими как поиск сообществ).
3. В качестве класса модели машинного обучения на графах наиболее оптимальным являются графовые нейронные сети.
4. Для начала обучения необходимо "перевести" граф в набор тензоров смежности/инцидентности и атрибутов аккаунтов и транзакций.
5. Библиотека PyTorch Geometric предоставляет наиболее известные реализации слоёв для графовых нейронных сетей - необходимо лишь структурированно (в требуемом формате) передать в них данные о графе.
6. Графовая нейронная сеть внимания показывает лучшие результаты по итогам обучения, поскольку она смогла идентифицировать большее число фишинговых аккаунтов.