УДК: 004.75

1.6. Обзор методов идентификации подозрительных адресов в публичных блокчейнах

Д. А. Зенюк
Институт прикладной математики им. М. В. Келдыша РАН, Москва

В работе дан обзор различных подходов к проблеме выявления подозрительных адресов в публичных блокчейнах с помощью методов машинного обучения, в первую очередь, методов классификации. Эта задача весьма актуальны в связи с тем, что все легальные участники рынка криптоактивов сейчас должны соблюдать достаточно строгие правила по уточнению источников средств, участвующих в любой обрабатываемой транзакции. Несмотря на то, что Bitcoin и подобные ему платежные системы считаются анонимными, алгоритмы, использующие последние достижениях в области машинного обучения и искусственного интеллекта вместе с тщательным подбором признаков, описывающих наблюдения, могут демонстрировать весьма хорошие результаты. Рассмотрение ведется в основном для сети Bitcoin, но отмечено несколько интересных примеров для Ethereum. Насколько можно судить, обзор такого рода публикуется на русском языке впервые.

Введение

Технологии распределенных реестров (*distributed ledgers*, DLT) активно развиваются и внедряются в финансовой индустрии последнее десятилетие. Созданные небольшой группой энтузиастов в попытке выстроить справедливую децентрализованную платежную инфраструктуру, свободную от излишней бюрократии и системной дискриминации, они относительно быстро завоевали популярность и достигли внушительной рыночной капитализации, что не позволяет относиться к ним как к нишевым решениям. Наиболее известными примерами являются Bitcoin и Ethereum, хотя современная экосистема криптоактивов насчитывает уже более 8 тыс. проектов (хотя многие из них и не представляют интереса). Блокчейн — это лишь одна из возможных архитектурных концепций DLT, но после триумфа Bitcoin именно ее обычно имеют в виду, говоря о распределенных реестрах, хотя это и не верно с позиции строгой терминологии. Настоящий текст посвящен одной группе задач, связанных именно с блокчейнами. Описание технических и инженерных аспектов различных моделей блокчейнов выходит далеко за рамки этой статьи и не нужно для понимания ее содержания. Заинтересованный читатель сможет найти больше подробностей, например, в [Narayanan, 2016; Tschorsch, 2016; Natarajan, 2017].

Блокчейны изначально были созданы так, чтобы доверие между участниками транзакции (которых далее будем называть также акторами) достигалось за счет использования криптографических алгоритмов. По этой причине в отличие от привычных банковских систем здесь не требуется идентификация отправителя и получателей — убедиться в правомерности сделки может любой на основе публично доступной информации. Акторы скрыты за псевдонимами, которые принято называть адресами. Создание новых адресов ничего не стоит и почти никак не ограничено. Поэтому один актор вполне может владеть (т.е. обладать приватным ключом, дающим право распоряжаться средствами) несколькими сотнями таких адресов-псевдонимов.

Однако быстро выяснилось, что именно по этим причинам блокчейны стали весьма популярны на сером и черном рынках. К примеру, в [Foley, 2019] объем транзакций, связанных с нелегальными сделками, в 2019 г. оценивался в 76 млн. долларов США, что составляло 46% от общего объема. Монетарные и фискальные власти достаточно быстро перешли к жесткой регуляции рынка криптоактивов. Сейчас все легальные игроки на этом рынке обязаны выполнять набор процедур по уточнению источников поступления средств. В силу этого весьма актуальной стала проблема разработки алгоритмических подходов к анализу истории транзакций, которые сейчас принято объединять под названием *know-your-transaction* (KYT). Важнейшую роль в таком анализе играют две тесно связанные задачи: кластеризация адресного пространства, которая позволяет с высокой степень уверенности выделить адреса, принадлежащие одному и тому же актору, и классификация адресов на «хорошие» и «плохие» (с регулятивной точки зрения). Связь задач объясняется тем, что если в кластере адресов хотя бы один или два являются плохими, то тогда и все остальные из осторожности также следует считать неблагонадежными, а транзакции от них отклонить. Другими словами, свойство быть хорошим или плохим характеризует акторов, а не отдельные адреса.

Для решения задачи кластеризации в отдельности было предложено множество т.н. эвристик. Формально их можно считать вырожденными решающими правилами, выполняющими кластеризацию на основе некоторого типичного поведенческого паттерна. Эвристики не предполагают настройки параметров в результате обучения. Несмотря на кажущуюся примитивность, они показали достаточную эффективность на практике. Более того, грубая эвристическая кластеризация часто служит первым этапом в построении более сложного инструментария, в которой уже используется современное машинное обучение. В настоящей статье будет дан обзор именно таких моделей, пригодных для автоматизации KYT-процедур.

Для дальнейшего изложениями нам понадобятся всего две эвристики: эвристика множественных входов (*multiple inpit* или *common spending*) и эвристика сдачи (*change address*). Обе они применяются только к блокчейнам с UTXO-моделью, к которым принадлежит, к примеру, Bitcoin. Первая эвристика основана на том, что если у транзакции несколько входов, то скорее всего, они принадлежат одному и тому же актору в силу особенностей процедуры подписи транзакции. Вторая эвристика связана с тем, что в UTXO-блокчейнах все входы транзакции должны быть потрачены полностью, из-за чего у отправителя часто возникает потребность переводить размен на какой-нибудь из своих адресов. Подробнее об эвристике множественных входов см. в [Harrigan, 2016], вторая была впервые подробно описана в [Meiklejohn, 2013].

Другой взгляд на проблему KYT-анализа можно получить с позиции теории графов. Наиболее полное описание истории транзакций в UTXO-блокчейнах дает граф адресов и транзакций (AT-граф). Он представляет собой двудольный ориентированный граф, в котором различаются вершины-адреса и вершины-транзакции. Ребра могут идти только от адресов отправителей к транзакции, а от нее — к адресам получателей. AT-графы похожи на сети Петри, и действительно, этот формализм был использован в [Pinna, 2018] для теоретического моделирования блокчейнов, однако с точки зрения KYT он не дает никакого особого преимущества. AT-граф заключает в себе всю доступную информацию о совокупности рассматриваемых транзакций, но в силу своей огромной размерности он обычно непригоден для непосредственного анализа и визуализации. Поэтому вместо него часто рассматривается граф адресов (A-граф). Его вершины отождествляются с адресами, а направленные ребра — с транзакциями. Вершина, из которой ребро исходит, соответствует отправителю, а та, в которою оно входит — получателю. В строгом смысле эти конструкции являются мультиграфами, поскольку одни и те же адреса могут участвовать в нескольких различных транзакциях, что порождает кратные ребра. A-графы могут быть построены для любого блокчейна, а не только для тех, которые используют модель UTXO. Как будет показано далее, графовые представления могут дать такие признаки для обучения, которые было бы слишком трудно вычленить сразу из одного только списка транзакций. Отметим здесь в заключение, что пока терминология для графовых представлений не стала общепризнанной, и поэтому, к примеру, то, что здесь называется A-графом, в других публикациях может называться сетью транзакций (*transaction network*) или как-нибудь еще.

Поскольку далее будет обсуждаться применение методов машинного обучения, остановимся на вопросе о нотации. Для удобства будем использовать короткие аббревиатуры для обозначения стандартных алгоритмов. Так, LR означает логистическую регрессию, RF — лес решающих деревьев, SVM — машину опорных векторов, kNN — метод k ближайших соседей, а ADB и XGB — соответственно мета-алгоритмы AdaBoost и XGBoost. Под RNN понимается reccurent neural net. Там, где используются resampling-алгоритмы для выравнивания диспропорции классов в обучающей выборке, RUS означает random undersampling, ROS — random oversampling, а SMOTE — synthetic minority oversampling technique. Хорошо известные термины (например, PCA или dropout) даются без пояснений. Термины и понятия, для которых пока нет общеупотребительных переводов на русский, выделены курсивом.

Методы

Достаточно большое количество работ предлагает использовать аппарат машинного обучения по прецедентам для создания KYT-инструментов, способных различать хороших и плохих акторов. Задача обычно ставится как проблема классификации, где необходимо присвоить каждому актору метку класса. В простейшем случае меток всего две: легальные акторы и злоумышленники, т.е. хорошие и плохие. Но известны модели с более богатым набором меток, например, дополнительно выделяют биржи, игорные сервисы, Darknet-площадки и т.п. В данном разделе описаны сами подходы к построению классификаторов и других моделей машинного обучения, а также их наиболее важные результаты. Метрики качества для тех моделей, для которых они были указаны авторами оригинальных работ, приведены в сводной таблице в конце стать.

Самой ранней работой по этой тематике, которую удается найти в открытом доступе, по всей видимости является [Yin, 2017]. Там приведено сравнение нескольких базовых алгоритмов классификации применительно к адресному пространству Bitcoin. В качестве исходных данных авторы использовали уже обработанные данные от Chainalysis (достаточно известной сейчас компании, которая занимается проведением KYT-расследований), т.е. адреса уже были кластеризованы и соотнесены с акторами, хотя подробности того, как это было сделано, не раскрываются. В размеченных данных было 874 кластера, в неразмеченных данных — 100 000 кластеров. Эти данные соответствуют истории 395 млн. транзакций. Для каждого адресного кластера (именно они трактовались как наблюдения в этом исследовании) известны различные числовые признаки, связанные со статистикой транзакций и характеристиками активности кластера. Итоговая размерность пространства признаков была равна 99. В размеченных данных используется 12 меток. Применялись обычные техники предварительной обработки (работа с пропущенными значениями, стандартизация и масштабирование и т.п.). В эксперименте участвовало 13 алгоритмов классификации: LR, SVM, RF, kNN, и некоторые другие. Хотя авторы и отмечают сильную диспропорцию классов, они не используют техники resampling. В [Harlev, 2018] и [Sun, 2019] исследован тот же самый набор данных: дизайн эксперимента в этих статья совпадает с [Yin, 2017], хотя некоторые детали, такие как выбор конкретных алгоритмов или использование resampling-методов, меняются в каждой статье.

В [Shao, 2018] для классификации адресов Bitcoin используется аппарат нейронных сетей. Всего авторами было собрано 8 986 адресов, сгруппированных в 66 уникальных кластеров, что соответствует 350 196 транзакциям, совершенным в период с января 2009 г. по сентябрь 2016. Та часть исходных данных, которая соответствует истории всех транзакций, в которых участвовал каждый адрес, затем была преобразована достаточно сложным образом. В начале авторы используют технику *one-hot encoding* (хотя не поясняют, каким дискретным признакам и атрибутам соответствует это кодирование), а затем, используя аналогию с задачами обработки естественных языков, получают эффективное представление этих закодированных данных в виде векторов фиксированной длины с помощью алгоритма word2vec. Затем к этому вектору присоединяют некоторые числовые характеристики транзакций (комиссии, сумма транзакции и т. п.), и подают на вход RNN слоя, чтобы получить векторное представление фиксированной длины для каждого адреса. Наконец, выполняется конкатенация этого вектора со стандартизованными и перемасштабированными статистическими признаками самого адреса, чтобы получить финальное 173-мерное представление. Этот 173-мерный вектор подается на вход трехслойной полносвязной нейронной сети с модифицированными ReLU активациями. В качестве функции эмпирического риска использовалась нестандартная конструкция, названная ими *additive margin softmax*.

Целый ряд работ посвящен использованию для классификации признаков, основанных на анализе графовых представлений транзакций. Например, в [Jourdan, 2018] авторы впервые предложили использовать для обучения данные о графовых мотивах. В строгом смысле, мотивами называются подграфы определенного вида, которые обладают структурной значимостью в контексте рассматриваемой задачи. Конкретно в этой работе *n*-мотивом называется путь в AT-графе длины 2*n*, содержащий *n* вершин-транзакций (обязательно уникальных) и *n* + 1 вершин-сущностей, которые вообще говоря могут повторяться. В их эксперименте *n* было не больше 3. Перед выделением мотивов часть адресов в AT-графе была предварительно «свернута» с помощью эвристики множественных входов. В качестве признаков рассматривались количество мотивов определенной формы, размер комиссий, входящие и выходящие потоки BTC, количество уникальных входов и выходов, индексы центральности и проч. Всего было выделено 315 признаков. В использованном наборе данных было 30 331 700 адресов, соответствующих 272 акторам, и 5 меток. Для классификации использовался градиентный бустинг для решающих деревьев (LighGBM) с выбором гиперпараметров на основе байесовской оптимизации.

Идею использования признаков, основанных на мотивах в AT-графе, развивает [Zola, 2019]. Используя данные о примерно 380 млн. транзакций между более чем 1 млрд. адресов (среди которых было выделено 311 акторов), авторы собрали 4 набора обучающих данных, описывающих акторов, адреса, а также 1- и 2-мотивы. Классификация проводилась с 6 метками, причем отмечена типичная для подобных задач диспропорция классов. Предлагается использовать stacking-классификатор (хотя в самой статье эту технику называют *cascading*). Сначала выполняется обучение трех независимых базовых классификаторов (были выбраны ADB, RF и XGB) по признакам адресов и мотивов, а затем результаты этих базовых классификаторов были объединены с набором исходных данных об акторах и обучен еще один финальный классификатор. Количество признаков в последнем классификаторе равно 25. В качестве алгоритмов для финальной классификации используется один из трех базовых. Для сравнения авторы также выполнили классификацию каждым алгоритмом на основе только лишь признаков самих акторов. Оказалось, что добавление признаков о мотивах позволило значительно увеличить F_1 метрику качества. Вычисление коэффициентов значимости (*importance scores*) признаков также подтвердило высокую значимость признаков на основе мотивов.

Еще одной работой, где используются графовые мотивы, является [Wu, 2021], посвященная идентификации адресов миксеров. Миксерами называются специальные сервисы, созданные для дополнительной обфускации движения средств между адресами. Они возникли как ответ на первые работы, показавшие, что Bitcoin вовсе не обеспечивает полной анонимности. В качестве исходных данных там рассматривалась история примерно 1 500 000 транзакций с ноября 2014 г. по январь 2016 г. Размеченные адреса составляли всего 0.19% от всех адресов, участвовавших в анализируемых транзакциях.

По полученным данным были построены A- и AT-графы. Авторы развивают концепцию ограниченных во времени мотивов (*temporal motifs*), т.е. они ищут только такие подграфы, в которых ребра возникают в одной и той же последовательности и могут иметь временные метки, различающиеся не более чем на заданную величину. Статистическую значимость мотивов вычисляли с помощью z-статистики, используя реальную частоту наблюдения мотива, а также ожидание и среднеквадратичное отклонение для частоты того же мотива в случайном графе (последние две величины оценивались по 100 реализациям специальной конфигурационной модели). Числовые признаки, которые использовались для классификации, были поделены на 3 группы. В первую попали 17 признаков, основанных на собранных статистических данных о мотивах и общих свойствах A-графа. Помимо этого, было еще 6 признаков для описания активности адресов, типа общего количества входящих и исходящих транзакций, а также их полные суммы и т.п. Наконец, к последней группе относились 6 признаков, описывающих роли адресов в т.н. транзакционных циклах (подробнее см. оригинальную статью), к примеру, среднее время между первой и последней транзакциями в цикле.

Задача рассматривалась как бинарная классификация. Чтобы преодолеть проблему крайне сильной диспропорции классов, авторы развивают свой оригинальный метод обучения. Коротко, их идея сводится к тому, чтобы сначала из неразмеченных данных выбрать объекты, которые почти наверняка не

относятся к миксерам, а затем с помощью этих объектов уже обучать классификатор. Для первого этапа используется обычная LR. На втором этапе используется еще одна LR со взвешенной по количеству объектов каждого класса функцией эмпирического риска. Эта обученная модель применяется ко всем оставшимся неразмеченным объектам.

Другой заслуживающий упоминания подход к поиску дополнительных признаков был описан в [Toyoda, 2019]. Там было предложено использовать для классификации признаки временной активности адресов. Целью работы была идентификация мошеннических HYIP (*high yield investment program*). Данные были собраны и размечены ими вручную. Эксперимент проводился в двух вариантах: в одном классификацию выполняли непосредственно по адресам и их признакам, во втором — адреса вначале были кластеризованы с помощью эвристики множественных входов, а признаки некоторым образом агрегированы. Независимо от схемы, для каждого адреса, подвергнутого анализу, из блокчейна извлекались все транзакции, в которых этот адрес выступал в качестве входа или выхода. Всего в размеченной части данных было 955 акторов. В качестве алгоритмов классификации авторы используют RF, XGB, SVM с RBF-ядром, kNN и очень простую трехслойную нейронную сеть с одним скрытым слоем. В [Toyoda, 2018] эти же авторы обсуждали почти ту же самую модель, но с большим количеством меток классов.

Идеи [Toyoda, 2019] развивает [Lin, 2019], добавив к базовым признакам характеристики активности адресов, под которыми понимается время между самой ранней и самой поздней транзакциями, где участвует адрес. Также вычисляются полное количество принятых и отправленных BTC, их долларовый эквивалент, среднее значение и среднеквадратичное отклонение баланса адреса (в BTC и USD). Наконец, к этому добавляются моменты, вычисленные по массивам характеристик, собранным за все время активности адреса. К примеру, если адрес участвует в 6 транзакциях, то массив будет содержать 6 наблюдений. Мотивация для использования моментов сводится к тому, что ожидание, дисперсия, асимметрия и эксцесс дают некоторую информацию о форме распределения (хотя в теории вероятностей хорошо известно, что моменты не всегда единственным образом определяют распределение). Оценка значимости признаков (по дополнительной информативности, *information gain*), показала, что из 10 наиболее важных признаков 6 были предложены именно в данной статье. Эксперимент использует 8 классификаторов, среди которых LR, SVM, ADB, RF, XGB, LightGBM и простая нейронная сеть (4 полносвязных слоя по 512 элементов с batch norm и dropout).

Выше уже было показано, что использование признаков, полученных с помощью вычислений на графах, позволяет зачастую улучшить качество классификации и выявить нетривиальные поведенческие паттерны, соответствующие нелегальной активности. Существует отдельная группа методов и техник — вложения графов (*graph embeddings*), — которая позволяет систематически и без ручной работы извлекать конечномерные векторные описания вершин в графах. Обзор этих методов выходит за рамки настоящего изложения, подробности см., например, в [Hamilton, 2017; Cai, 2018; Goyal, 2018] и цитированной там литературе. Такой подход оказался весьма плодотворен при решении рассматриваемых задач классификации адресов и акторов.

В [Hu, 2019] приведено сравнение результатов, которые можно получить с помощью рандомизированных методов вложения графов. В качестве референсных были взяты модели с обычными признаками, подготовленными вручную (всего 14 признаков). Задача рассматривалась как бинарная классификация, где нужно было выделить подозрительные транзакции (этим работа отличается от всех остальных, где классификация выполнялась для адресов или акторов). Авторы сравнили 4 модели: графовый аналог kNN, классификатор ADB по базовым признакам, классификатор ADB на основе объединенного описания из базовых признаков и числовых признаков графовых вложений, и, наконец, ансамбли, где в качестве базовых рассматривались классификаторы отдельно по «ручным» и графовым признакам. В качестве базового классификатора для ансамблей использовался ADB с одним решающим деревом. Всего в графе было 37 907 769 обычных транзакций и 7 461 895 связанных с отмыванием денег, что вместе составляло всего 27.3% от всех зафиксированных за этот срок транзакций. Первые две модели показали весьма посредственное качество. Наивный графовый kNN, к примеру, имеет F_1 всего 0.09, что неприемлемо мало. Классификатор на основе графовых вложений оказался заметно лучше. Ансамбль позволил еще немного улучшить качество классификации. В дополнение к этому авторы провели интересный дополнительный тест обобщающей способности: они изучили, насколько хорошо тестируемые модели могут различать транзакции от миксеров, которые не были предъявлены во время обучения, подробности см. в оригинальной статье.

Похожая методика была опробована в [Liang, 2019] для выявления адресов крупных бирж. Задача также рассматривалась как бинарная классификация. Исходные данные — 1 356 519 транзакций с участием 3 100 148 уникальных адресов. Судя по тексту, эвристики кластеризации не использовались. Всего была выделена 121 торговая площадка с 69 224 адресами, которые участвовали в 89 085 транзакциях. Для снижения диспропорции меток применялся RUS. Для графового вложения использовалась рандомизированная техника DeepWalk. Классификация выполнялась с помощью 5 моделей: примитивная нейронная сеть с одним скрытым слоем, линейный SVM, LR, RF и обычное решающее дерево.

В [Weber, 2019] были приведены результаты первого применения сетей с графовой сверткой (*graph convolution network*, GCN) к проблеме классификации адресов в сети Bitcoin. В качестве референсных моделей рассматривались LR, RF и простая нейронная сеть с одним внутренним слоем. Поскольку дизайн эксперимента предполагал изучение динамики A-графов, также была апробирована

специализированная архитектура EvolveGCN. В [Alarab, 2020] отталкиваются от работы [Weber, 2019], предлагая несколько видоизмененную архитектуру самой нейронной сети — она включает в себя дополнительные слои и использует другую нормализацию матрицы смежности (это мотивируется тем, что изначально GCN были созданы для неориентированных графов с симметрической матрицей Лапласа, а графы транзакций ориентированны).

Наконец, в [Oliveira, 2021] развивают методику [Weber, 2019] с помощью дополнительных признаков, полученных с помощью специального случайного блуждания по графу транзакций, которые было названо ими GuiltyWalker. Суть метода достаточно проста: начиная с какой-нибудь вершины (из заданного списка) случайным образом с равной вероятностью выбирается одну из предшествующих вершин (т.е. блуждание распространяется против направлений ребер), продолжая до тех пор, пока либо не останется подходящих вершин, либо пока не встретится вершина, имеющая метку плохой. Блуждание развивается назад во времени, от более поздних к более ранним транзакциям. В силу природы данных, у любого блуждания всегда будет терминальная точка, т.е. бесконечное блуждание невозможно. Успешным считается блуждание, которое окончилось на нелегальной транзакции. Требуемое число таких успехов является гиперпараметром. Поскольку не от всякой транзакции можно, двигаясь назад, достичь нелегальной, то авторы специально в начале работы определяют вершины, для которых успех возможен, с помощью вспомогательных методов обхода графов. Траектории моделируются до тех пор, пока не будет получено заданное количество успешных.

На основе результатов моделирования набора таких выборочных траекторий вычисляются новые признаки транзакций. Среди них дескриптивные статистики длины траекторий, доля успешных траекторий от общего числа траекторий, испущенных из данной вершины, и количество уникальных терминальных вершин. Дизайн эксперимента такой же, как и [Weber, 2019]. В качестве референса использовалась классификация RF с фиксированными гиперпараметрами. Вычисления повторялись для исходного набора признаков, для признаков только из GuiltyWalker и для объединенного описания. Чтобы еще улучшить результат, был проведен отбор признаков с помощью перестановочного теста (*permutation importance*), и 5 наиболее значимых признаков, выделенных с помощью GuiltyWalker, были объединены с исходными. Именно эта схема в итоге показала лучший результат. Авторам действительно удалось немного улучшить результат [Weber, 2019], от которой они отталкивались.

В [Pocher, 2022] было приведено сравнение современных методов анализа графов нейронными сетями со стандартными методами машинного обучения. В сравнении участвовали LR, RF, SVM, kNN, ADB, GCN и еще одна интересная архитектура — *graph attention network* (GAT). Архитектура GAT использует механизм внимания, который до этого помог добиться существенного продвижения в задачах машинного перевода. В описанном эксперименте лучший результат показала GCN, в то время как GAT оказалось лишь на третьем месте с результатом, сравнимым с обычным решающим деревом. Тем не менее, эта работа, насколько можно судить, является первым примером использования архитектуры GAT для задач KYT-исследований, и, как отмечают сами авторы, результат вполне возможно будет лучше при использовании большего количества внутренних слоев (в самой статье был всего один специальный графовый слой).

Всюду выше речь шла только о задачах классификации. Еще одна группа методов для поиска нелегальный транзакций и плохих адресов, которая относится уже к обучению без учителя — это поиск аномалий и выбросов. Основная идея заключается в том, что нелегальные транзакции должны по ряду признаков значительно отличаться от обычных легальных транзакций. Если представить себе транзакции как точки в многомерном пространстве признаков, то такие атипичные транзакции должны соответствовать разреженной периферии, далеко отстоящей от плотных скоплений нормальных транзакций. Для поиска аномалий используются самые различные техники: варианты *local outlier factor* (LOF), SVM с одним классом, изолирующие леса и эллиптические огибающие. Учитывая тесную связь KYT-анализа с графами, упомянем также о том, что существует особый набор техник для поиска аномалий в графах, хотя не удалось найти примеры их использования именно для поиска подозрительных транзакций, адресов и акторов в блокчейнах.

Более систематическое сравнение нескольких методов выявления аномалий дано в [Pham, 2016]. В качестве данных для своего эксперимента они использовали историю транзакций в сети Bitcoin вплоть до 7 апреля 2013 г., что соответствует 6 336 769 акторов и 37 450 461 транзакций. Признаки были выделены вручную и включали в себя степени и полустепени вершин, средние времен между транзакциями, балансы адресов, время активности и несколько других. Для выявления аномалий использовался метод k-средних (обучение выполнялось лишь по нескольким базовым признакам) с $k = 7$, по результатам которого опционально вычислялся LOF. Тем не менее, из 30 недавних на тот момент времени (2016 г.) краж их метод смог выявить только одну релевантную аномалию.

В [Monamo, 2016] было приведено исследование методики поиска аномалий с помощью предварительной кластеризации. Было выделено 14 признаков, собранных в 3 группы: дескриптивные статистики посланных и принятых сумм BTC, характеристики вершин в A-графе (степени вершин, коэффициенты кластеризации, количество треугольников с данной вершиной) и характеристики локального окружения вершин в нем же. Сравнивались два алгоритма кластеризации: привычный алгоритм k-средних и его робастная модификация, более устойчивая к иррегулярным формам кластеров. Сразу же следует оговориться, что кластеризация используется только как промежуточный этап для поиска аномалий. Этот

подход достаточно спорен, что, впрочем, признают и сами авторы. Количество кластеров (т.е. параметр *k*) определялось с помощью прямого перебора и оценки внутрикластерного рассеяния. В итоге было выбрано *k* = 8. В качестве аномалий в обоих случаях рассматривались объекты, которые были наиболее удалены от соответствующего центроида (т.е. дизайн эксперимента в целом очень напоминает предыдущую работу). Результаты эксперимента оказались неоднозначными. Так, из 30 известных случаев кражи BTC алгоритм смог выявить в лучшем случае 5.

В [Prado-Romero, 2018] был описан алгоритм для выявления адресов Bitcoin-миксеров с помощью поиска аномалий. В основе лежала простая идея о том, что миксер в силу специфики своей работы должен нарушать естественное разбиение E-графа на сообщества, поскольку миксер связывает транзакциями акторов, которые в привычных для них обстоятельствах вряд ли бы взаимодействовали. На первом этапе к A-графу после использования эвристики множественных входов применяется лувенский алгоритм (*Louvain method for community detection*), хотя, разумеется, любой другой современный алгоритм поиска сообществ тоже подошел бы. Затем внутри каждого сообщества подсчитывается усредненное расстояние между вершинами. Показатель аномальности для каждой вершины *v* теперь вычисляется как количество таких пар вершин (*v*, *u*), где *u* принадлежит тому же сообществу *C*, что и *v*, для которых расстояние между ними превосходит усредненное значение для *C*. Чтобы протестировать свой подход, авторы собрали два небольших (примерно по месяцу) фрагмента из истории транзакций за 2012 и 2013 гг. В качестве контрольной группы они рассматривали всего 6 адресов, которые скорее всего принадлежали миксерам по результатам исследования [Möser, 2013].

В [Nan, 2018] для решения той же самой задачи поиска адресов миксеров в сочетании с алгоритмами поиска аномалий была использована техника вложения графов. Здесь апелляция к поиску аномалий была обоснована тем, что миксер занимает особенное положение в графе транзакций: он служит точкой сопряжения или мостом между несколькими несвязанными между собой сообществами адресов, принадлежащих конечным пользователям сервиса. Авторы утверждают, что этот вывод был основан на анализе нескольких известных миксеров, хотя данные [Möser, 2013], где был описан масштабный эксперимент по изучению устройства миксеров, не подтверждает эту гипотезу полностью. Там действительно часто можно найти вершину, играющую роль моста, но неясно, исчерпывает ли это все адреса, принадлежащие миксеру и активно используемые им в работе. Для эффективного представления структуры графа использовался подход на основе популярной архитектуры *encoder—decoder*. После получения векторных представлений вершин к ним применялся обычный метод k-средних с *k* = 15. Наконец, по результатам этой вспомогательной кластеризации вычисляются значения *local outlier probability* (LoOP) и вершины с наибольшими значениями этого параметра объявляются миксерами. Для предварительной кластеризации использовалась эвристика множественных входов. Оказалось, что вычисленные значения LoOP для 4 известным им адресов миксеров были заметно выше чем среднее значение LoOP по всем адресам: не менее чем 0.81 против 0.32.

Тут следует отметить, что в отличие от обучения по прецедентам, для работ, использовавших кластеризацию и поиск аномалий, крайне трудно адекватно оценить точность и эффективность. Если такой анализ вообще проводился, то он был ситуативным и подходил только для данных, описанных в конкретной статье. Вообще, сама концепция поиска нелегальной активности через выявления аномалий и выбросов может быть не совсем верна, поскольку потенциальные злоумышленники наоборот стремятся сделать свои действия незаметными на фоне других транзакций, имитировать нормальное поведение. К примеру, в [Lorenz, 2020] была обнаружено именно такая картина: известные нелегальные объекты в выборке были плотно окружены нормальными транзакциями.

В заключение, в силу нехватки места, лишь кратко отметим еще несколько работы, содержащих интересные подходы. Так, [Lorenz, 2020] использует парадигму активного обучения, т.е. в изначальном наборе данных, в котором размечена лишь небольшая часть объектов (или нет разметки вовсе), на основе некоторого алгоритма выделяется порция объектов, которые предлагаются для изучения и разметки аналитику. Эти новые объекты используются для переобучения вспомогательной прецедентной модели, и процедура повторяется до тех пор, пока не будут достигнуто удовлетворительные качество на некоторой отложенной выборке. В [Remy, 2018] был предложен оригинальный способ улучшения кластеризации адресов на основе поиска сообществ в графах.

Все упомянутые работы были посвящены сети Bitcoin, что, впрочем, вполне понятно, учитывая долю в совокупной капитализации всех криптоактивов. Тем не менее, для второй по значимости сети, Ethereum, исследования такого рода тоже проводились. В ней используется модель транзакций, отличная от UTXO, и популярная эвристика множественных входов неприменима. Следует отметить работу [Poursafaei, 2020]. Там в исходных данных было 53 087 адресов, участвовавших в 18 686 447 транзакций. Каждый объект в обучающей выборке характеризуется 54 признаками, которые условно разделены на группы: общие признаки (баланс, время активности), соседство (степени соответствующих вершин в A-графе), локальные признаки (дескриптивная статистика типа минимум, максимум, среднее и среднеквадратичное отклонение по количеству входящих и исходящих транзакций и суммам) и временные характеристики (дескриптивная статистика по временам меду последовательными транзакциями). К этим признакам применялся метод PCA для ортогонализации в пространстве признаков, чтобы устранить возможную коллинеарность. Классификация понималась в бинарном смысле (хороший или плохой). Поскольку классы сильно диспропорциональны — известных плохих адресов значительно меньше, —

также были опробованы разные методы resampling. Использовались следующие алгоритмы: LR, линейная SVM, ADB и RF. В [Chen, 2018] была предложена модель для выявления мошеннических схем в сети Ethereum. Важную роль там играли признаки, полученные по статистике движения эфира (*ether*) — специального ресурса, играющего важную роль в функционировании смарт-контрактов. В [Tam, 2019] архитектура GCN с некоторыми дополнительными техниками была, среди прочего, применена к данным из сети Ethereum, показав достаточно хорошее качество прогноза в классификации с 7 метками.

Заключение

Здесь собраны данные по качеству прогноза моделей из упоминавшихся статей, где обучение выполнялось по данным с разметкой. Сразу же отметим, что прямое сравнение не всегда корректно, поскольку дизайн экспериментов очень сильно различался от одной публикации к другой. Если явно не оговорено обратное, то числа в таблице означают F_1 метрику. Она представляет собой геометрическое среднее точности (*precision*) и чувствительности (*recall*, *true positive rate*, TPR). Иногда используется также *false positve rate* (FPR). Нужно помнить, что для задач, где было больше двух классов, есть разные способы усреднения этих метрик (*macro-average* и *micro-average*). Здесь это различие не отмечено, поскольку в некоторых работах приводят оба значения, а в некоторых — только одно, причем без всяких пояснений о методе усреднения. В таблицу внесены только несколько лучших моделей, если в статье было опробовано их несколько.

Yin, Vatrupa, 2017	XGB 0.8056
Chen et al., 2018	XGB 0.86
Harlev et al., 2018	RF 0.67, bagging 0.72, SMOTE+boosting 0.76
Jourdan et al., 2018	LightGBM 0.91
Toyoda, Mathiopoulos, Ohtsuki, 2019	RF 0.9433 (TPR), 0.0641 (FPR)
Weber et al., 2019	RF 0.759, GCN 0.628, EvolveGCN 0.72
Hu et al., 2019	ADB 0.94, ensemble 0.95
Liang et al., 2019	RF 0.91
Lin et al., 2019	LightGBM 0.87
Tam et al., 2019	LR 0.652, boosting 0.6995
Zola et al., 2019	RF 0.98, XGB 0.9968
Alarab, Prakoonwit, Nacer, 2020	GCN 0.773
Lorenz et al., 2020	XGB 0.76, RF 0.83
Poursafaei, Hamad, Zilic, 2020	ROS+RF 0.995, ROS+ADB 0.998
Oliveira et al., 2021	RF 0.85
Wu et al., 2021	оригинальный алгоритм 0.7874 (TPR), 0.0991 (FPR)
Pocher et al., 2021	GAT 0.723, RF 0.782, GCN 0.844

Таблица: Оценка качества прогноза в разных моделях

Основной вывод, который можно сделать из этой таблицы — проблема выявления акторов с плохой репутацией, которые скорее всего замешаны в противоправном движении криптоактивов, и принадлежащих им адресов вполне может быть решена на должном уровне качества при использовании тщательно подобранных моделей. Особенно перспективными здесь выглядят глубокие нейронные сети для графов, в частности, различные специализированные GCN. Основным препятствием является не концептуальная сложность задачи, а скорее нехватка качественно размеченных данных — почти всем авторам приходилось использовать либо уже готовые разметки небольшого размера, либо использовать свой собственные *ad hoc* инструменты.

Развитием инструментов для выполнения KYT-анализа сейчас вполне успешно занимаются многие компании. Помимо упомянутой в самом начале Chainalysis, это CipherTrace, Elliptic, Bitfury (проект Crystal), IdentityMind, Scorechain, Traceer, Blockseer. Развитие алгоритмических подходов к классификации адресов и акторов в сочетании с другими специализированными методами цифровой криминалистики (*digital forensics*) вполне могут стать решающим шагом на пути к институционализации рынка криптоактивов и превращения его в полноценную часть мировой финансовой системы.

Литература

1. Alarab I., Prakoonwit S., Nacer M. I. Competence of graph convolutional networks for anti-money laundering in Bitcoin blockchain // Proceedings of the 2020 5th International conference on machine learning technologies. – 2020. – С. 23-27.
2. Cai H., Zheng V. W., Chang K. C. C. A comprehensive survey of graph embedding: Problems, techniques, and applications // IEEE transactions on knowledge and data engineering. – 2018. – Т. 30. – №. 9. – С. 1616-1637.
3. Chen W. et al. Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology // Proceedings of the 2018 World Wide Web conference. – 2018. – С. 1409-1418.

4. Foley S., Karlsen J. R., Putniņš T. J. Sex, drugs, and bitcoin: How much illegal activity is financed through cryptocurrencies? // The Review of Financial Studies. – 2019. – Т. 32. – №. 5. – С. 1798-1853.
5. Goyal P., Ferrara E. Graph embedding techniques, applications, and performance: a survey // Knowledge-Based Systems. – 2018. – Т. 151. – С. 78-94.
6. Hamilton W. L., Ying R., Leskovec J. Representation learning on graphs: Methods and applications // arXiv:1709.05584. 2017.
7. Harlev M. A. et al. Breaking Bad: De-anonymising entity types on the Bitcoin blockchain using supervised machine learning // The 51st Hawaii International Conference on System Sciences. HICSS 2018. – 2018. – С. 3497-3506.
8. Harrigan M., Fretter C. The unreasonable effectiveness of address clustering // 2016 Intl IEEE conferences on ubiquitous intelligence & computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people, and smart world congress. – 2016. – С. 368-373.
9. Hu Y. et al. Characterizing and detecting money laundering activities on the Bitcoin network // arXiv:1912.12060. – 2019.
10. Jourdan M. et al. Characterizing entities in the Bitcoin blockchain // 2018 IEEE International conference on data mining workshops (ICDMW). – 2018. – С. 55-62.
11. Liang J. et al. Bitcoin exchange addresses identification and its application in online drug trading regulation. // 23rd Pacific Asia Conference on Information Systems: Secure ICT Platform for the 4th Industrial Revolution, PACIS 2019. – 2019.
12. Lin Y. J. et al. An evaluation of Bitcoin address classification based on transaction history summarization // 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). – 2019. – С. 302-310.
13. Lorenz J. et al. Machine learning methods to detect money laundering in the Bitcoin blockchain in the presence of label scarcity // Proceedings of the first ACM international conference on AI in finance. – 2020. – С. 1-8.
14. Meiklejohn S. et al. A fistful of Bitcoins: characterizing payments among men with no names // Proceedings of the 2013 conference on Internet measurement conference. – 2013. – С. 127-140.
15. Monamo P., Marivate V., Twala B. Unsupervised learning for robust Bitcoin fraud detection //2016 Information Security for South Africa (ISSA). – 2016. – С. 129-134.
16. Möser M., Böhme R., Breuker D. An inquiry into money laundering tools in the Bitcoin ecosystem // 2013 APWG eCrime researchers summit. – 2013. – С. 1-14.
17. Nan L., Tao D. Bitcoin mixing detection using deep autoencoder // 2018 IEEE Third international conference on data science in cyberspace (DSC). – 2018. – С. 280-287.
18. Narayanan A. et al. Bitcoin and cryptocurrency technologies: a comprehensive introduction. – Princeton University Press, 2016.
19. Natarajan H., Krause S., Gradstein H. Distributed ledger technology and blockchain. FinTech Note No. 1. – Washington, DC: World Bank, 2017.
20. Oliveira C. et al. GuiltyWalker: Distance to illicit nodes in the Bitcoin network // arXiv:2102.05373. – 2021.
21. Pham T., Lee S. Anomaly detection in the Bitcoin system – a network perspective // arXiv:1611.03942. – 2016.
22. Pinna A. et al. A Petri nets model for blockchain analysis // The Computer Journal. – 2018. – Т. 61. – №. 9. – С. 1374-1388.
23. Pocher N. et al. Detecting anomalous cryptocurrency transactions: an AML/CFT application of machine learning-based forensics // arXiv:2206.04803. – 2022.
24. Poursafaei F., Hamad G. B., Zilic Z. Detecting malicious Ethereum entities via application of machine learning classification // 2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS). – 2020. – С. 120-127.
25. Prado-Romero M. A., Doerr C., Gago-Alonso A. Discovering Bitcoin mixing using anomaly detection // Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaíso, Chile, Proceedings 22. – Springer International Publishing, 2018. – С. 534-541.
26. Remy C., Rym B., Matthieu L. Tracking Bitcoin users activity using community detection on a network of weak signals // Complex Networks & Their Applications VI: Proceedings of Complex Networks 2017. – Springer International Publishing, 2018. – С. 166-177.
27. Shao W. et al. Identifying Bitcoin users using deep neural network // Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018, Guangzhou, China, Proceedings, Part IV 18. – Springer International Publishing, 2018. – С. 178-192.
28. Sun Yin H. H. et al. Regulating cryptocurrencies: a supervised machine learning approach to de-anonymizing the Bitcoin blockchain // Journal of Management Information Systems. – 2019. – Т. 36. – №. 1. – С. 37-73.

29. Tam D. S. H. et al. Identifying illicit accounts in large scale E-payment networks – A graph representation learning approach // arXiv:1906.05546. – 2019.
30. Toyoda K., Mathiopoulos P. T., Ohtsuki T. A novel methodology for HYIP operators' Bitcoin addresses identification // IEEE Access. – 2019. – Т. 7. – С. 74835-74848.
31. Toyoda K., Ohtsuki T., Mathiopoulos P. T. Multi-class Bitcoin-enabled service identification based on transaction history summarization // 2018 IEEE international conference on Internet of things, green computing and communications, social computing and smart data. – 2018. – С. 1153-1160.
32. Tschorsch F., Scheuermann B. Bitcoin and beyond: A technical survey on decentralized digital currencies // IEEE Communications Surveys & Tutorials. – 2016. – Т. 18. – №. 3. – С. 2084-2123.
33. Weber M. et al. Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics // arXiv:1908.02591. – 2019.
34. Wu J. et al. Detecting mixing services via mining Bitcoin transaction network with hybrid motifs // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2021. – Т. 52. – №. 4. – С. 2237-2249.
35. Yin H. S., Vatrapu R. A first estimation of the proportion of cybercriminal entities in the Bitcoin ecosystem using supervised machine learning // 2017 IEEE international conference on big data (Big Data). – 2017. – С. 3690-3699.
36. Zola F. et al. Cascading machine learning to attack bitcoin anonymity // 2019 IEEE International Conference on Blockchain. – 2019. – С. 10-17.

Д. А. Зенюк
Институт прикладной математики им. М. В. Келдыша РАН, Москва
eldrich@yandex.ru

Ключевые слова: блокчейн, криптоактивы, анонимность, классификация, кластеризация, графовые свертки.

Dmitry A. Zenyuk, Identification of Suspicious Addresses in Public Blockchains: a Survey

Keywords
blockchain, cryptoassets, anonymity, classification, clusterization, graph convolutions.

DOI: 10.34706/DE-2024-03-06

JEL classification: C65, E42

Abstract
The paper surveys techniques for identification of potentially malicious addresses in public blockchains based on machine learning, foremost, classification methods. This problem is especially important now, when all legal platforms must abide to strict rules and verify sources of every processed transaction. Despite seeming anonymity of Bitcoin and similar systems, algorithms based on recent advances in machine learning and AI with thorough feature selection demonstrate quite good quality. Exposition is mainly given for Bitcoin network, but several interesting examples for Ethereum are also mentioned.