

УДК 519.7

Клионкин В.С.

Студент магистратуры

Самарского национального исследовательского университета

имени академика С.П. Королева

Институт информатики и кибернетики

Кафедра технической кибернетики

г. Самара, РФ

Научный руководитель: Белоусов А.А.

Доцент кафедры технической кибернетики, к.ф.-м.н.

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ ФИКТИВНЫХ БИРЖЕВЫХ СДЕЛОК

Аннотация

Объектом исследования является автоматическое определение фиктивности сделок на криптовалютных биржах.

Цель работы – изучить применимость методов анализа временных рядов для задачи обнаружения фиктивных сделок, применить эти методы на практике и произвести их улучшение для достижения наиболее приемлемых результатов обнаружения фиктивных сделок.

В процессе работы были разработаны алгоритм по выявлению фиктивных сделок и соответствующая программа, позволяющая пользователю получать данные о сделках, а также обучены классификаторы и искусственная нейронная сеть, выявляющие фиктивные сделки. Система позволяет получение данных о сделках и выявление их фиктивности.

Были изучены три модели, позволяющие прогнозировать фиктивность сделок, а на основе полученных данных была разработана собственная модель машинного обучения. Были проведены эксперименты и исследования выбранных методов анализа и обнаружения фиктивных сделок, сравнение и анализ результатов. Сделано обобщение полученных результатов. Подведено заключение в работе.

Ключевые слова

Фиктивные сделки, биржевая торговля, криптовалюты, нейросети, автоэнкодер, машинное обучение, временные ряды, обнаружение аномалий, классификация.

Введение

В современном мире криптовалюты становятся все более привычными. Для них создаются свои сервисы, многие магазины начинают принимать их к оплате, а многие уже начинают считать криптовалюты чем-то обычно-повседневным.

В мире уже произошло много событий, которые свидетельствуют о принятии криптовалют в качестве альтернативного класса активов. К примеру, недавно операции с криптовалютой провела компания Tesla, купив \$1,5 млрд биткоинов в феврале и продав 10% цифровых монет в конце марта. Автопроизводителю удалось зафиксировать прибыль в \$101 млн, и благодаря этому побить рекорд выручки за первый квартал. В течение 2022 года из-за наложенных на Россию санкций многие россияне начали активно пользоваться криптовалютами как средством обхода санкционных ограничений, осуществления торговых операций и вывода денег из страны.

На фоне этого многие криптовалюты любыми способами пытаются увеличить объем свой торгов, чтобы получить максимальную прибыль. Один из таких способов – wash trading, т.е. фиктивные сделки. Фиктивные сделки негативно влияют на рынок криптовалют и доверие к криптовалютам и криптовалютным биржам, и в связи с этим биржи активно ищут способы определения подобных

операций.

Определение фиктивной сделки

Фиктивная сделка (wash trading) — процесс, при котором трейдер покупает и продает один и тот же актив с явной целью предоставления рынку вводящей в заблуждение информации [1]. Совокупность фиктивных сделок можно представить как позицию группы манипуляторов, где позиция — это количество активов, принадлежащих трейдеру. Участвующий в фиктивной торговле трейдер стремится сохранить свою собственную позицию неизменной для минимизации ненужных финансовых потерь, и, следовательно, позиция всей группы фиктивной торговли также не меняется. В процессе фиктивных сделок позиция меняется из-за размещения манипуляторами новых ордеров, и её можно представить как:

$$P + O \rightarrow P,$$

где P — позиция, O — новые ордера. Позиция состоит из множества ордеров:

$$P = \{O_1, O_2 \dots O_n\},$$

где P — позиция, O — новые ордера. Каждый ордер можно представить как:

$$O = (Tr, T, P, V),$$

где O — ордер, Tr — идентификатор трейдера, T — тип (покупка/продажа), P — цена, V — объём. Если рассматривать покупку и продажу как положительное и отрицательное изменение позиции, то можно добавить соответствующий знак к Tr и V и избавиться от необходимости указывать тип сделки. Теперь ордер будет выглядеть так:

$$O = (\pm Tr, P, \pm V).$$

Ордера из таблицы 3 можно представить так:

$$P = \{(-A, 121, -500), (B, 122, 510), (-B, 120.5, -510), (A, 121.8, 500)\}$$

Сопоставленные пары ордеров будут выглядеть так:

$$P = \{(B - A, 122, 510 - 500 = 10), (A - B, 121.8, 500 - 510 = -10)\}$$

Разность между ценами, по которым закрываются ордера, является интервалом между ценами транзакций. Цена транзакции 122 имеет разность $122 - 121 = 1.0$, а у цены 121.8 разность равна $121.8 - 120.5 = 1.3$. При соединении транзакций, интервалы цен которых пересекаются, получится новое представление позиции:

$$P = \{(B - A + A - B, 121.25 \pm 0.75, 10 - 10)\} = \{0, 121.25 \pm 0.75, 0\}$$

Идентификатор трейдера «0» означает, что оба участника сговора участвовали и в покупке, и в продаже активов, а нулевой объём означает, что их транзакции взаимно перекрываются, и активы не были по-настоящему проданы или куплены. Таким образом, неизменная позиция с взаимно перекрывающимися идентификаторами трейдеров и нулевым объёмом указывает на факт сговора и осуществления фиктивных сделок.

Однако такое простое правило даёт сбой в случае участия в схеме нескольких трейдеров, поскольку большое количество участников фиктивной торговли даёт большое разнообразие комбинаций из торгующих между собой пар трейдеров. Чтобы определить сложные схемы из нескольких участников, был создан алгоритм. Представим множество ордеров, поданных трейдерами, в виде направленного графа, где вершины являются трейдерами, а рёбра — ордерами. Рассмотрим пример фиктивной сделки с несколькими сторонами в потоке нормальных транзакций, приведённый в таблице 1:

Таблица 1

Пример сложной фиктивной сделки в потоке нормальных транзакций

№	Трейдер	Время	Тип сделки	Цена	Объём
1	А	06:01:001	Продажа	121.00	550
2	В	06:01:002	Покупка	121.02	600
3	В	06:05:000	Продажа	120.95	600

№	Треjder	Время	Тип сделки	Цена	Объём
4	C	06:05:001	Покупка	121.01	550
5	F	06:11:021	Продажа	121.30	120
6	C	06:14:003	Покупка	121.30	140
7	C	06:35:003	Продажа	121.00	550
8	D	06:35:004	Покупка	121.01	600
9	C	06:41:000	Продажа	120.75	350
10	E	06:57:000	Покупка	120.90	400
11	D	07:03:047	Продажа	121.01	550
12	A	07:03:048	Покупка	121.01	550
13	E	07:55:001	Продажа	121.80	500
14	F	07:56:008	Покупка	121.80	500

Среди 14 ордеров, поданных 6 трейдерами, четыре пары ордеров размещены в малом временном промежутке по не выбивающейся из прочих цене, а их объёмы приблизительно совпадают, так что ордера в каждой паре подозрительно легко сопоставить и закрыть.

На рисунке 1 показана схема закрытия ордеров из таблицы 1; закрытия ордеров показаны рёбрами графа, выделенными пунктиром. Из приведённой схемы видно, что действия трейдеров, участвующих в фиктивной торговле, образуют замкнутый цикл. В результате замыкания цикла можно увидеть, что финансовая выгода прошла по всей цепочке трейдеров и вернулась к первому участнику, то есть позиции участвующих в сговоре трейдеров фактически не изменились.

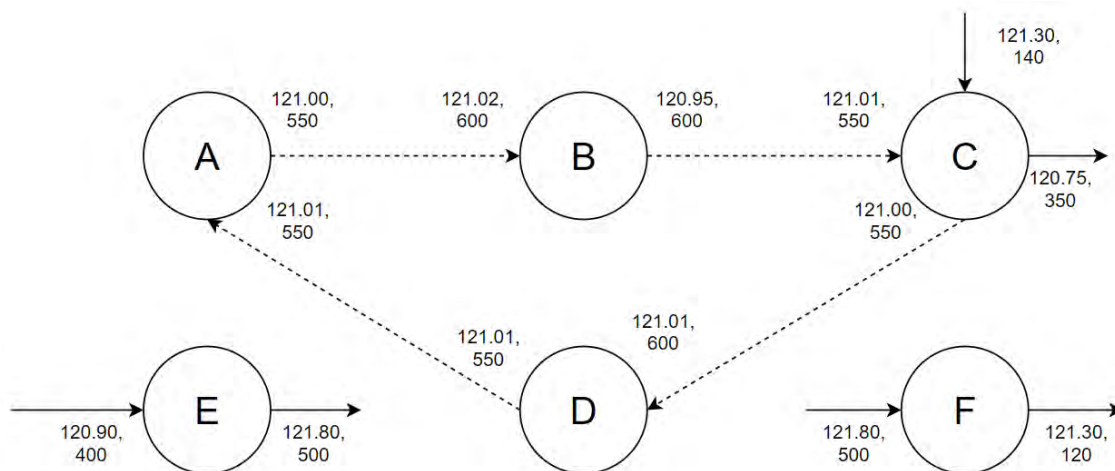


Рисунок 1 – Схема закрытия ордеров и сговора трейдеров

Из вышеприведённого анализа следует, что ведение фиктивной торговли имеет две отличительные черты:

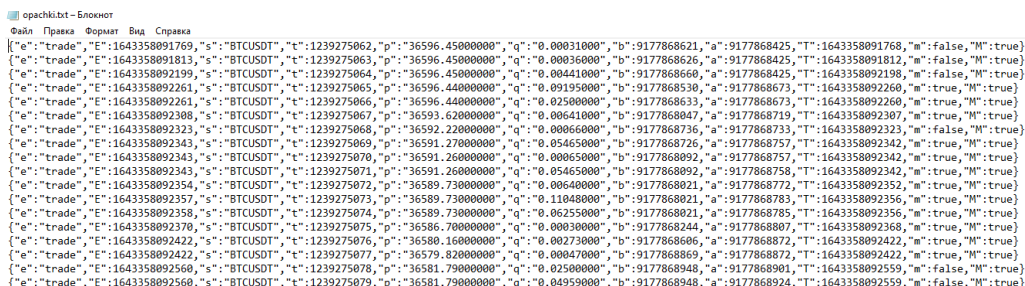
- совпадающие ордера – трейдеры намеренно размещают совпадающие ордера друг за другом через минимальные промежутки времени, чтобы гарантировать их подбор и закрытие;
- замкнутый цикл транзакций – закрытие совпадающих ордеров являются фиктивной сделкой только если эти ордера являются частью замкнутого цикла транзакций.

В примере из таблицы 1 манипуляторы размещают ордера цикла 1 и 2 в 6:01, но фиктивная сделка не завершается полностью до размещения ордера 12, который закрывает цикл транзакций, в 7:03. Следовательно, фиктивная сделка может быть обнаружена с помощью определения совпадающих пар

ордеров и замкнутого цикла. Сначала находятся совпадающие пары ордеров по правилу сопоставления, а затем среди этих пар находятся те, которые составляют замкнутый цикл.

Сбор и обработка данных

Для анализа данных и выбора способа классификации был создан датасет, состоящий из реальных записей о транзакциях с криптовалютной биржи Binance [2]. С помощью разработанного модуля подключения к серверу-источнику данных по протоколу WebSocket была записана история транзакций одного торгового дня. На рисунке 2 показан вид файла, в который были сохранены собранные данные.



```
opachkitit - Блокнот
Файл  Правка  Формат  Вид  Справка
[{"e": "trade", "E": "1643358091769", "s": "BTCUSDT", "t": "1239275862", "p": "36596.45000000", "q": "0.00031000", "b": "9177868621", "a": "9177868425", "T": "1643358091768", "m": false, "M": true},
{"e": "trade", "E": "1643358091813", "s": "BTCUSDT", "t": "1239275863", "p": "36596.45000000", "q": "0.00036000", "b": "9177868626", "a": "9177868425", "T": "1643358091812", "m": false, "M": true},
{"e": "trade", "E": "1643358092199", "s": "BTCUSDT", "t": "1239275864", "p": "36596.45000000", "q": "0.00441000", "b": "9177868660", "a": "9177868425", "T": "1643358092198", "m": false, "M": true},
{"e": "trade", "E": "1643358092261", "s": "BTCUSDT", "t": "1239275865", "p": "36596.44000000", "q": "0.09195000", "b": "9177868630", "a": "9177868673", "T": "1643358092260", "m": true, "M": true},
{"e": "trade", "E": "1643358092261", "s": "BTCUSDT", "t": "1239275866", "p": "36596.44000000", "q": "0.02500000", "b": "9177868633", "a": "9177868673", "T": "1643358092260", "m": true, "M": true},
{"e": "trade", "E": "1643358092308", "s": "BTCUSDT", "t": "1239275867", "p": "36593.62000000", "q": "0.00641000", "b": "9177868047", "a": "9177868719", "T": "1643358092307", "m": true, "M": true},
{"e": "trade", "E": "1643358092323", "s": "BTCUSDT", "t": "1239275868", "p": "36592.22000000", "q": "0.00066000", "b": "9177868736", "a": "9177868733", "T": "1643358092323", "m": false, "M": true},
{"e": "trade", "E": "1643358092343", "s": "BTCUSDT", "t": "1239275869", "p": "36591.27000000", "q": "0.05465000", "b": "9177868726", "a": "9177868757", "T": "1643358092342", "m": true, "M": true},
{"e": "trade", "E": "1643358092343", "s": "BTCUSDT", "t": "1239275870", "p": "36591.26000000", "q": "0.00065000", "b": "9177868092", "a": "9177868757", "T": "1643358092342", "m": true, "M": true},
{"e": "trade", "E": "1643358092343", "s": "BTCUSDT", "t": "1239275871", "p": "36591.26000000", "q": "0.05465000", "b": "9177868092", "a": "9177868758", "T": "1643358092342", "m": true, "M": true},
{"e": "trade", "E": "1643358092354", "s": "BTCUSDT", "t": "1239275872", "p": "36589.73000000", "q": "0.00640000", "b": "9177868021", "a": "9177868772", "T": "1643358092352", "m": true, "M": true},
{"e": "trade", "E": "1643358092357", "s": "BTCUSDT", "t": "1239275873", "p": "36589.73000000", "q": "0.11048000", "b": "9177868021", "a": "9177868783", "T": "1643358092356", "m": true, "M": true},
{"e": "trade", "E": "1643358092358", "s": "BTCUSDT", "t": "1239275874", "p": "36589.73000000", "q": "0.06255000", "b": "9177868021", "a": "9177868785", "T": "1643358092356", "m": true, "M": true},
{"e": "trade", "E": "1643358092370", "s": "BTCUSDT", "t": "1239275875", "p": "36586.70000000", "q": "0.00030000", "b": "9177868244", "a": "9177868807", "T": "1643358092368", "m": true, "M": true},
{"e": "trade", "E": "1643358092422", "s": "BTCUSDT", "t": "1239275876", "p": "36580.16000000", "q": "0.00273000", "b": "9177868606", "a": "9177868872", "T": "1643358092422", "m": true, "M": true},
{"e": "trade", "E": "1643358092422", "s": "BTCUSDT", "t": "1239275877", "p": "36579.82000000", "q": "0.00047000", "b": "9177868869", "a": "9177868872", "T": "1643358092422", "m": true, "M": true},
{"e": "trade", "E": "1643358092508", "s": "BTCUSDT", "t": "1239275878", "p": "36581.79000000", "q": "0.02500000", "b": "9177868948", "a": "9177868901", "T": "1643358092559", "m": false, "M": true},
{"e": "trade", "E": "1643358092560", "s": "BTCUSDT", "t": "1239275879", "p": "36581.79000000", "q": "0.04959000", "b": "9177868948", "a": "9177868924", "T": "1643358092559", "m": false, "M": true}]
```

Рисунок 2 – Собранные данные о транзакциях

Эти записи хранят в себе следующие данные: время совершения сделки, идентификаторы участников сделки с обеих сторон, цена и объём сделки и прочую информацию, не являющуюся важной для классификации (актив, с которым совершается сделка, является ли покупатель маркет-мейкером и пр.).

После сбора эти данные были хронологически отсортированы, после чего размечены по алгоритму, описанному в разделе 1.3: для каждой транзакции был произведён поиск цикла, и если такой цикл был найден, транзакция размечалась как фиктивная, в противном случае – как настоящая (см. рисунок 3).

	e	E	s	t	p	q	b	a	T	m	M	wash
0	trade	1668163703851	BTCUSDT	2160142894	17330.0	0.00123	15395661274	15395661239	1668163703850	False	True	1
1	trade	1668163703851	BTCUSDT	2160142895	17330.58	0.00574	15395661274	15395661237	1668163703850	False	True	1
2	trade	1668163703851	BTCUSDT	2160142896	17330.65	0.00665	15395661274	15395661174	1668163703850	False	True	0
3	trade	1668163703851	BTCUSDT	2160142897	17330.99	0.00418	15395661274	15395661097	1668163703850	False	True	0
4	trade	1668163703869	BTCUSDT	2160142898	17330.59	0.01897	15395661286	15395661276	1668163703869	False	True	0

Рисунок 3 – Размеченные данные о транзакциях

После разметки данных были проведены нормализация численных признаков и кодирование категориальных признаков в численные.

Анализ позиции трейдеров

Фиктивные транзакции невозможно обнаружить путём классификации данных о транзакциях обычными алгоритмами машинного обучения, поскольку активы продаются по ценам и объёмам, соответствующим рыночным. Однако при проведённом анализе фиктивных сделок была выявлена следующая характерная черта – изменения позиции на некотором временном промежутке стремятся к 0, т.е. в конце временного промежутка позиция будет крайне незначительно отличаться от начальной.

После сбора и разметки данных были построены временные ряды для позиций каждого трейдера. Анализ данных о позициях подтвердил правоту данного вывода – у трейдеров, которые участвовали в фиктивных сделках, позиция, несмотря на одно или несколько скачкообразных изменений, изменялась к концу временного промежутка на малые величины. В противовес этому, у настоящих трейдеров позиция претерпевала к концу промежутка значительные изменения. На рисунках 4 и 5 приведены примеры позиции трейдера-участника фиктивных сделок и обычного трейдера соответственно.

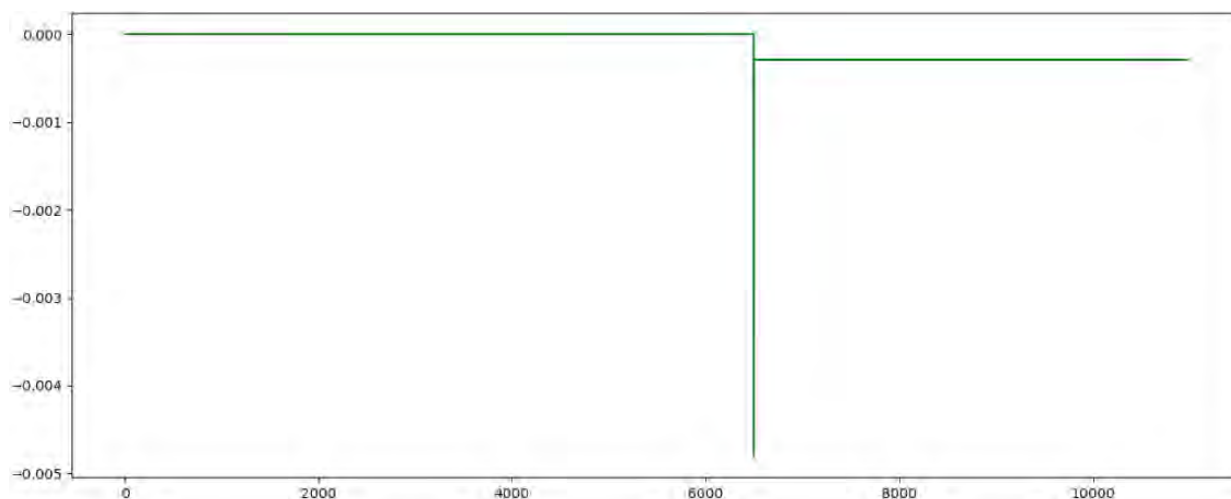


Рисунок 4 – Позиция трейдера-участника фиктивных сделок

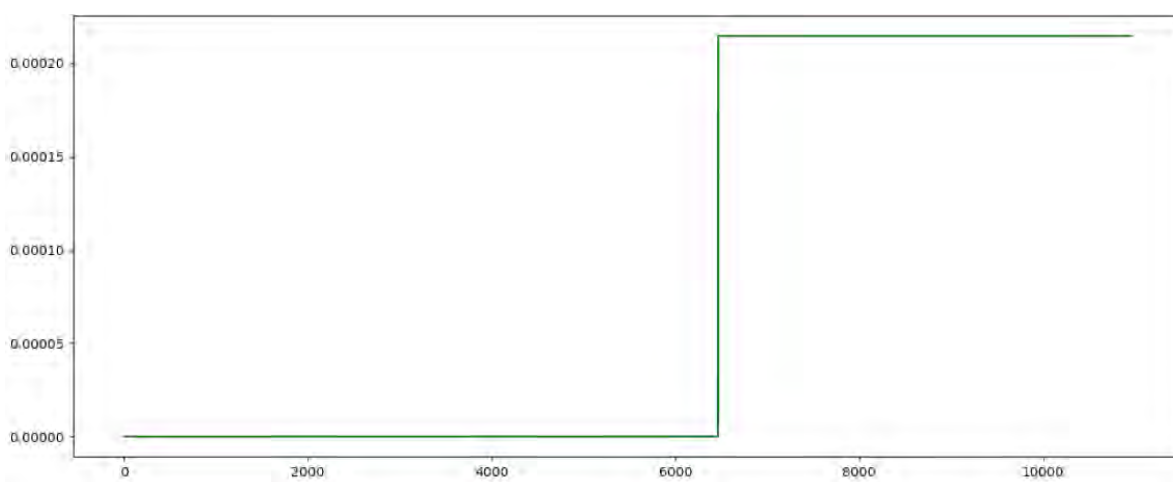


Рисунок 5 – Позиция обычного трейдера

Для того, чтобы отличить настоящие сделки от фиктивных, необходимо построить для каждого трейдера временной ряд изменения его позиции. Если он не соответствует нормальному ряду, то, значит, он состоит из фиктивных сделок.

Методы машинного обучения в задаче обнаружения аномалий

Данная задача относится к задачам обнаружения аномалий во временных рядах. Обнаружением аномалий (выбросов) называется задача обнаружения объектов, не похожих на остальные данные, не подчиняющихся ожидаемым, «нормальным» паттернам поведения [3].

Методы машинного обучения широко применяются в задачах обнаружения аномалий, потому что в большинстве практических случаев требуется обнаруживать аномалии в огромных объемах данных, из которых невозможно вручную извлечь статистически значимую информацию. В таких случаях машинное обучение помогает упростить процесс обнаружения аномалий и сэкономить ресурсы. К тому же методы машинного обучения показывают отличные результаты, когда в задачах задействованы большие наборы данных.

Одноклассовый метод опорных векторов

Метод опорных векторов (support vector machines) – алгоритм обучения с учителем, который используется для задач классификации и регрессионного анализа [4]. Одноклассовый метод опорных векторов (One-class SVM) использует только один класс, так как набор обучения содержит только примеры из целевого класса. Он выводит, какие свойства являются нормальными для объекта в целевом классе, а

затем на основе таких свойств прогнозирует, какие примеры отличаются от нормальных примеров. Это полезно для обнаружения аномалий, так как нехватка примеров для обучения является неотъемлемым свойством аномалий.

Изолирующий лес

Алгоритм изолирующего леса (iForest) был первоначально предложен Фэй Тони Лю, Кай Мин Тином и Чжи-Хуа Чжоу в 2008 году. В 2010 году было разработано расширение алгоритма - SCiforest для устранения кластеризованных аномалий и аномалий, параллельных оси. В 2012 году те же авторы продемонстрировали, что iForest имеет линейную временную сложность, небольшие требования к памяти и применим к данным большой размерности. Изолирующий лес разделяет пространство признаков и присваивает более высокие оценки аномалий тем данным, для изоляции которых требуется меньше разделений [5].

Автоэнкодеры

Автоэнкодер — это нейронная сеть прямого распространения, которая восстанавливает входной сигнал на выходе [6]. Внутри у неё имеется скрытый слой, который представляет собой код, описывающий модель. На рисунке 6 показано строение нейросети-автоэнкодера.

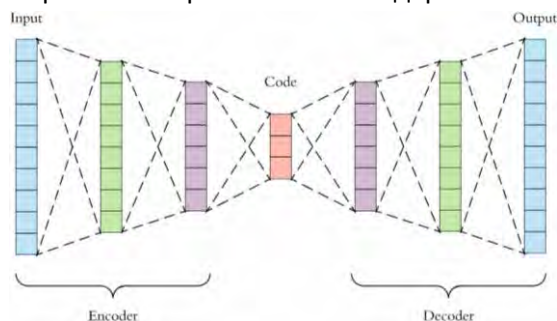


Рисунок 6 – Схема работы нейросети-автоэнкодера

Автоэнкодер состоит из двух частей: кодировщика, который представлен функцией $h = g(x)$, которая переводит входной сигнал в его представление (код), и декодировщика, который восстанавливает сигнал из его представления, используя функцию $x = f(h)$. Нейросеть, изменяя функции кодирования и декодирования, пытается выучить тождественную функцию $x = f(g(x))$, чтобы свести к минимуму функцию ошибки $L(x, f(g(x)))$.

Автоэнкодеры применяются в том числе и для задачи детектирования аномалий. Если в обучающей выборке аномальных объектов значительно меньше, чем обычных, то для хранения информации об обычных объектах потребуется меньше информации, чем для запоминания аномальных. В таком случае, если подобрать правильные параметры нейронной сети, она сможет запоминать и восстанавливать только обычные объекты, так как на аномальные ей не будет хватать обобщающей способности. Поэтому восстановленные моделью данные будут значительно отличаться от исходных.

Создание и обучение модели

Модель автоэнкодера, построенная в данной работе, состоит из двух связанных между собой полносвязных нейронных сетей. Нейросеть-кодировщик состоит из четырёх полносвязных слоёв, имеющих функцию активации LeakyReLU. Данная функция была выбрана, чтобы предотвратить «умирание» нейронов из-за отрицательных входных значений, т.к. позиция может изменяться и в отрицательную сторону. Входной слой имеет размерность, соответствующую размеру кодируемого временного ряда, а последующие слои осуществляют кодирование этой последовательности всё меньшим количеством параметров вплоть до 32. Нейросеть-декодировщик имеет размерность входного слоя 32, а последующие слои воссоздают всё большее количество параметров вплоть до изначального размера временного ряда. Для создания модели была использована библиотека для построения глубоких нейронных сетей PyTorch.

Для обучения модели была подготовлена выборка данных, состоящая из 12000 записей (по числу трейдеров) об изменениях позиции за сутки. Выборка данных для обучения была разделена на три части: тренировочную, тестовую и валидационную в пропорции 3-1-1.

После разделения выборка данных была преобразована в формат Dataset, используемый моделями PyTorch для доступа к данным. Далее был создан объект нейросети, которая описывается созданной моделью. Затем было проведено обучение модели: в каждой эпохе обучения рассчитывался выходной сигнал, считалось значение ошибки и проводилось её обратное распространение. На рисунке 74 показан график потерь при обучении в зависимости от количества итераций для функции потерь MSE (mean squared error, среднеквадратическая ошибка).

Далее на основе разностей между реальными и восстановленными рядами был обучен классификатор, который и определяет, являются ли сделки, составляющие эту позицию, фиктивными, или же нет.

Вычислительные эксперименты

Для оценки качества классификаторов в данной работе была выбрана метрика F1-score, т.к. в данной задаче одинаково важны точность и полнота классификатора [7].

После обучения каждого тип классификатора была проведена проверка их качества на реальных данных. Было выбрано 1000 случайных записей о транзакциях, после чего каждый классификатор предсказал, к какому классу (настоящая – 0, фиктивная – 1) относится каждая транзакция.

Одноклассовый метод опорных векторов не справился с задачей классификации позиций с фиктивными сделками. Лучшее значение F1-меры, которое данный классификатор показал при различных сочетаниях гиперпараметров, составляет 0,17.

Изолирующий лес также не справился с задачей классификации позиций с фиктивными сделками. Лучшее значение F1-меры, которое данный классификатор показал при различных сочетаниях гиперпараметров, составляет 0,16.

Было обучено несколько вариантов созданной модели-автоэнкодера. Было проведено обучение с разным количеством эпох (20, 50 и 100) и разными функциями потерь (квадратичная ошибка и функция Хьюбера).

По результатам обучения и тестирования модели стало ясно, что при слишком большом количестве эпох (100) она переобучается и становится непригодна для классификации реальных данных. Лучшие результаты показывает модель, обученная за 50 эпох с функцией Хьюбера в качестве функции потерь.

В таблице 2 приведено сравнение метрики общей точности и метрик точности, полноты и F1-меры для целевого класса для различных видов классификаторов.

Таблица 2

Сравнение метрик для разных типов классификаторов

Метрика	Изолирующий лес	Одноклассовый SVM	Автоэнкодер + классификатор
Accuracy	0.19	0.33	0.96
Precision	0.13	0.27	0.83
Recall	0.07	0.15	0.83
F1-score	0.09	0.19	0.83

По результатам сравнения качества моделей было выявлено, что только сочетание модели-автоэнкодера и классификатора позволило с высокой точность предсказывать, состоит ли позиция из фиктивных сделок. Такие алгоритмы, как одноклассовый метод опорных векторов и изолирующий лес оказались неспособны классифицировать фиктивные сделки.

Заключение

В процессе выполнения выпускной работы был разработан метод распознавания фиктивных сделок на криптовалютных биржах. Сущность используемого метода распознавания состоит в классификации

результата воссоздания истории позиции трейдера моделью-автоэнкодером.

В ходе выполнения работы были приведены основные понятия и определения предметной области, проведён анализ процесса осуществления фиктивных сделок и дано его математическое определение, был сформулирован способ выявления фиктивных сделок путём определения аномальных временных рядов, осуществлена математическая постановка задачи, проведён обзор методов выявления аномальных временных рядов.

Были проведены эксперименты по обучению классификаторов, которые применяются в задачах обнаружения аномалий, модели-автоэнкодера собственноручно созданной архитектуры. В результате экспериментов было выявлено, что только классификация результатов модели-автоэнкодера справляется с обнаружением временных последовательностей позиций фиктивных сделок.

Дальнейшее развитие работы можно продолжить в направлении обнаружения других типов аномалий в последовательностях сделок и усложнения модели – использования других признаков для обнаружения аномалий.

Список использованной литературы:

- 1 Blockchain Media [Электронный ресурс]. URL: <https://blockchain-media.org/what-is-wash-trading/> (дата обращения: 17.11.2021).
- 2 Binance Spot API Docs [Электронный ресурс]. URL: <https://github.com/binance/binance-spot-api-docs/blob/master/web-socket-streams.md> (дата обращения: 15.03.2023).
- 3 Соболев К.В. Автоматический поиск аномалий во временных рядах: дис. магистра. М., 2018. 12 с.
- 4 SVM. Подробный разбор метода опорных векторов [Электронный ресурс]. URL: <https://habr.com/ru/companies/ods/articles/484148/> (дата обращения: 01.03.2023).
- 5 Выявление мошенничества с помощью алгоритмов случайного леса, нейронного автокодировщика и изолирующего леса [Электронный ресурс]. URL: <https://habr.com/ru/companies/nix/articles/478286/> (дата обращения: 01.03.2023).
- 6 Автоэнкодеры: типы архитектур и применение [Электронный ресурс] URL: <https://neurohive.io/ru/osnovy-data-science/avtojenkoder-tipy-arhitektur-i-primeneniye/> (дата обращения: 01.03.2023).
- 7 Метрики классификации и регрессии [Электронный ресурс] URL: <https://academy.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii> (дата обращения: 17.03.2023).

© Клионкин В.С., 2023