



Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математической статистики

Сороколетова Ольга Евгеньевна

Прогнозирование временных рядов рынка криптовалют

Выпускная квалификационная работа

Научный руководитель:

к.ф.-м.н.

Т.В. Захарова

Москва, 2019

Содержание

1	Введение	3
2	Постановка задачи	5
2.1	Представление в виде задачи классификации	5
2.2	Нестационарность	5
3	Метод случайного отбора (RSM)	7
3.1	Основная идея	7
3.2	Схема отбора	8
3.3	Кодировщик	9
3.4	Архитектура LSTM-блока	9
4	Набор данных	12
5	Предобработка данных	13
5.1	Входные данные	13
5.2	Выходные данные	13
6	Применение моделей	14
6.1	Настройка гиперпараметров	14
6.2	Результаты тестирования	14
6.3	Альтернативная схема отбора	15
6.4	Проверка универсальности	16
7	Заключение	17
	Список литературы	18

1 Введение

Многие научные публикации в области компьютерных наук, а также экономические и финансовые статьи недавних лет посвящены адаптации алгоритмов машинного обучения к проблеме прогнозирования финансовых временных рядов. Однако в существующей литературе есть пробел, касающийся неисследованной области применения методов машинного обучения к временным рядам с данными о торговле криптовалютой.

Применение методов глубокого обучения сталкивается с необходимостью поиска сложного компромисса: с одной стороны, глубокие модели требуют большого количества данных для обучения, на практике имея в виду высокочастотные данные с не более, чем минутным, временным интервалом, с другой же, на длительном временном интервале далеко не всегда приходится наблюдать стационарность, и, следовательно, извлечение конкретных признаков, шаблонов и знаний о структуре варьируется в зависимости от сегмента обучающей выборки.

Вопрос применимости глубоко обучения к прогнозированию рынка все еще находится под обсуждением. В статьях [15], [17], [18] с помощью алгоритмов глубоко обучения были получены некоторые эмпирические результаты, показавшие, что, возможно, существует универсальность в формировании рыночных цен. В вышеуперечисленных работах было использовано огромное количество специальным образом упорядоченных данных для обучения рекуррентных нейросетей.

Метод случайного отбора (RSM) – это основанный на обучении метрике метод, который представляет собой другой подход. В нем по схеме отбора, описанной в дальнейших разделах, устанавливаются эталонные экземпляры, а далее оценивается схожесть между входным и установленными экземплярами. Метка наиболее похожего из примеров становится кандидатом на выход предсказательной модели.

Появление метода тесно связано с недавними разработками глубокого обучения в области обработки изображений. В статье [10] затрагиваются те же проблемы ограниченности количества релевантных тренировочных данных и отсутствия знаний о подходящих для введения метрики схожести трансформациях признаков пространства. Авторами (Hilliard, Nathan, Lawrence Phillips, Scott Howland) предпринимается попытка заменить стандартный подход с использованием статичного метрического сравнения на тренировку нейросети, способной обучаться самостоятельному сравнению с шаблоном последователь-

ностей, принадлежащих к разным классам. Результаты в [10] показывают, что подобная гибкая архитектура обеспечивает высокую эффективность в задаче классификации изображений.

Обучение метрике было реализовано ранее также в работах [13] и [14]. Задачей же данной дипломной работы становится адаптирование алгоритма к задаче анализа временных рядов. Идейной основой послужило предположение о том, что этот инструмент способен справиться с высокой нестационарной динамикой цифровых активов, в особенности же, таких нестабильных, как криптовалюта (в частности, биткоин).

Лучшие из современных алгоритмов глубоко обучения для анализа временных рядов, такие, как сети долгой краткосрочной памяти (LSTM), требуют огромных наборов данных для тренировки и поэтому страдают от того, что структурные признаки во временных рядах из криптовалюты могут существенно измениться на тестовой выборке по сравнению с тренировочной, результатом чего становятся неэффективность прогнозирования, выучивание шумов и противоречивость выходных данных. Например, в случае биткоина недавние данные являются более релевантными для предсказания тренда, чем поздние, что на практике ограничивает число экземпляров каждого класса. Адаптированный RSM, в свою очередь, представляет собой метод, в котором алгоритм выявляет лучшие недавние паттерны перед выставлением метки, достигая тем самым оптимальности предсказаний.

Итак, рассмотрим, как метод случайного отбора применяется для решения задачи прогнозирования временных рядов рынка криптовалют, которые отличаются нестационарностью, и сравним эффективность данного подхода с двумя базовыми методами глубокого обучения: нейросетями прямого распространения (MLP) и рекуррентными LSTM-нейросетями, обратив внимание на проблему дисбаланса классов, которая может быть сглажена применением RSM.

Покажем также, что эффективность классификации посредством RSM, выраженная в виде F-меры, значительно превышает шансы случайного процесса, доказывая, что извлечение определенной структурной информации для классификации направлений движения цен, а следовательно, прогнозирования рынка, в некоторой степени возможно.

2 Постановка задачи

2.1 Представление в виде задачи классификации

Формальная постановка задачи предсказания движения цены акции на рынке может быть сформулирована следующим образом.

Имеется три возможных варианта движения цены после временной отметки t : вверх (up), вниз (down) или без изменений (static). Будем рассматривать распределение логарифмической доходности R_t :

$$R_t = \log\left(\frac{P_t}{P_{t-1}}\right), \quad (1)$$

где P_t — значение цены в момент времени t , взятое из временного ряда. Обозначим через $p_t^{(up)}$, $p_t^{(down)}$ и $p_t^{(static)}$ вероятности наступления соответствующих верхним индексам событий. Эти вероятности будут получены в результате обучения модели. Таким образом, возникает задача трехклассовой классификации:

$$X_t \in \{up, down, static\}. \quad (2)$$

2.2 Нестационарность

В данном случае рассматривается ситуация, где $p_t^{(up)}$, $p_t^{(down)}$ и $p_t^{(static)}$ изменяются как функции времени t . С точки зрения алгоритмов машинного обучения, может случиться, что модели, обученные на таком наборе данных, будут смещены в сторону определенного класса и не будут справляться с данной проблемой классового дисбаланса корректно. Однако существует возможность сгладить эту проблематичную ситуацию с помощью walk-forward-оптимизации ([5], рисунок 1), которая заключается в том, что модель тренируется на подвыборке внутри тренировочного окна, тестируется в тестовом окне, а затем производится сдвиг обоих окон вперед, чтобы начать тренироваться снова. Этот метод позволяет использовать предположение о стационарности, что делает обучение более стабильным. Однако, если размер тренировочной выборки сильно ограничен, может страдать обобщающая способность моделей глубокого обучения.

Более формально суть алгоритма заключается в следующем. Будем считать, что если две последовательности $(x_{t-1}, x_{t-2}, \dots, x_{t-T-1})$ и $(x_{i-1}, x_{i-2}, \dots, x_{i-T-1})$ близки в смысле

некоторой метрики, то распределения $p_t^{(up)}$, $p_t^{(down)}$, $p_t^{(static)}$ и $p_i^{(up)}$, $p_i^{(down)}$, $p_i^{(static)}$ также близки.

Таким образом, задача предсказания метки движения цены сводится к тому, чтобы обучить модель корректно устанавливать метрику сравнения последовательностей.

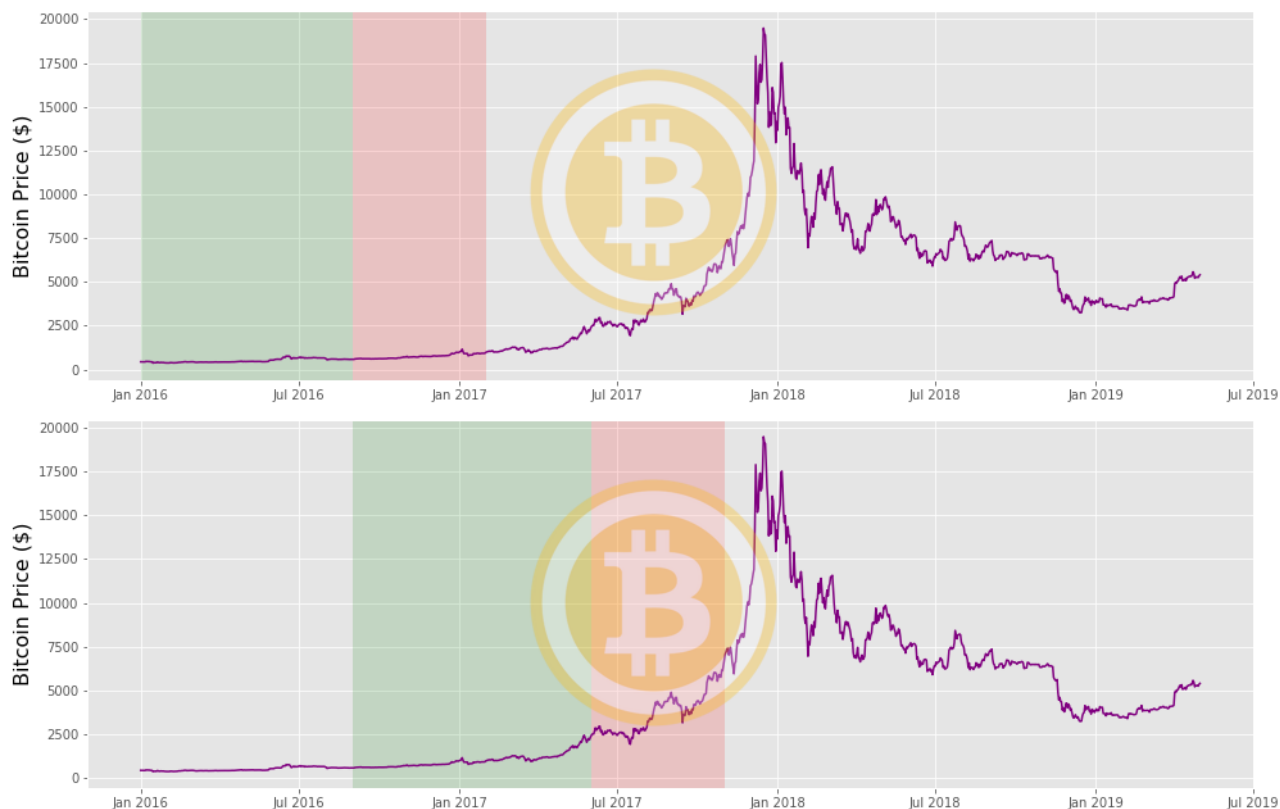


Рис. 1: Визуализация walk-forward-оптимизации. Зеленая область – тренировочное окно, красная область – тестовое окно.

3 Метод случайного отбора (RSM)

Раздел посвящен детальному рассмотрению схемы отбора и архитектуры модели в сравнении с общеизвестными архитектурами нейросетей, применяющимися при решении задачи прогнозирования временных рядов. На рисунке 2 представлена общая схема метода.

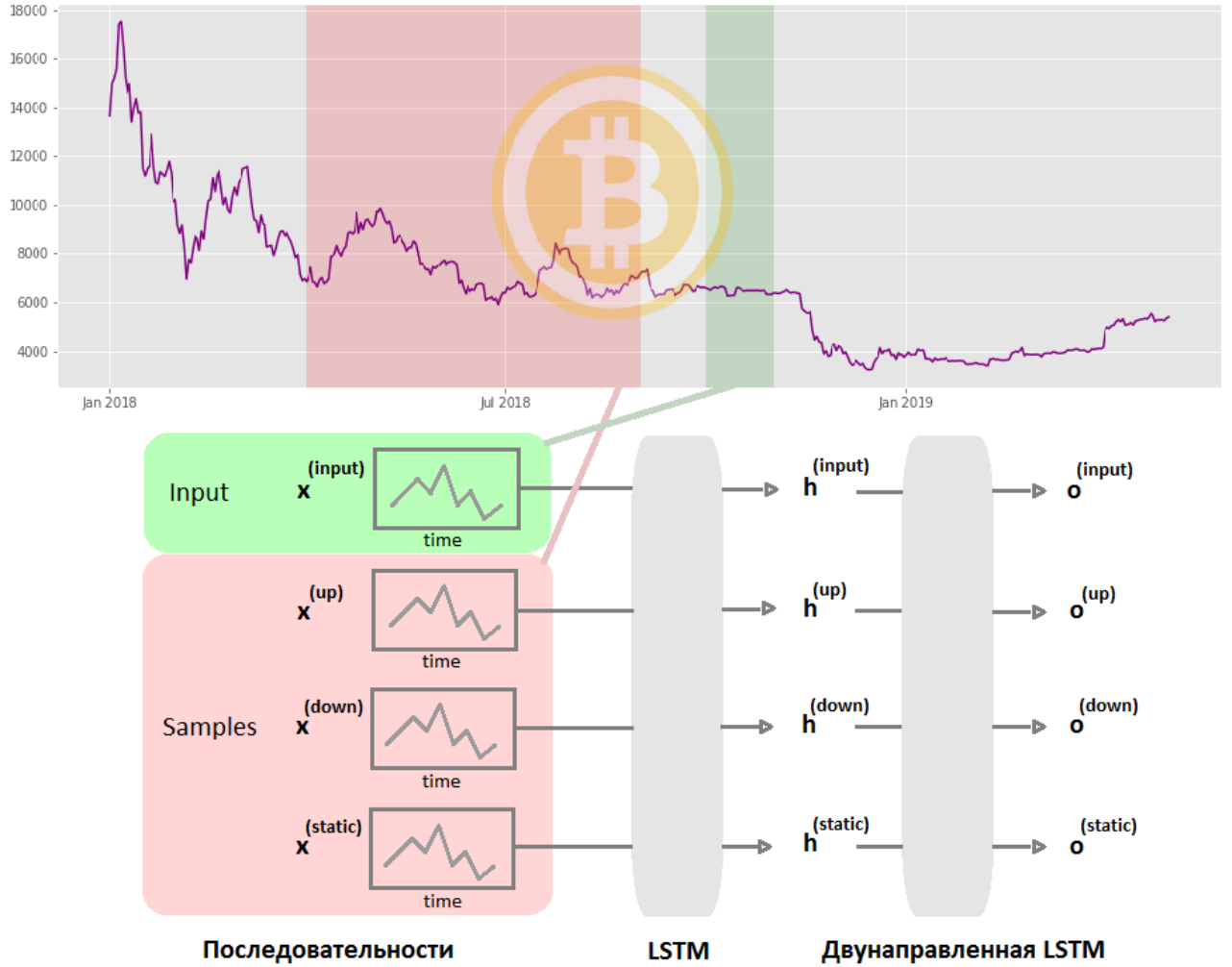


Рис. 2: Дана входная последовательность $x_t^{(input)}$, последовательности $x_t^{(up)}$, $x_t^{(down)}$ и $x_t^{(static)}$ случайно выбираются из красного окна $[t - k - l, t - k]$. Затем входная и выбранные последовательности независимо проходят через кодировщик LSTMNet и двунаправленную Bi LSTMNet (детально далее).

3.1 Основная идея

В основе метода лежит предположение о том, что схожесть пары последовательностей может быть охарактеризована при помощи классов, к которым они принадлежат: после-

довательность с меткой «up» более близка последовательности с меткой «up», чем последовательности с меткой «down» или «static».

На вход модели подаются пары элементов, один из которых – это последовательность, подлежащая классификации, а другой – набор последовательностей (три последовательности, помеченные как «up», «down» и «static» соответственно), выбранных из недавнего прошлого. Затем осуществляется получение внутренних представлений путем независимого кодирования каждой последовательности и сравнение внутреннего представления входной последовательности и внутренних представлений выбранных последовательностей. Выходным значением является класс, к которому предположительно принадлежит целевая последовательность.

Оптимизация параметров модели производится таким образом, чтобы достигались внутренние представления, для которых верно: среди внутренних представлений последовательностей из тройки более близко внутреннему представлению целевой последовательности, то, которое имеет метку класса, совпадающую с истинной в данном случае меткой. Близость внутренних представлений оценивается при помощи косинусной меры сходства.

3.2 Схема отбора

Схема отбора – это алгоритм, по которому осуществляется выбор тройки эталонных последовательностей, подающейся затем на вход модели. Она построена, основываясь на предположении, что финансовые данные нестационарны, что обуславливает необходимость выбора последовательностей исключительно среди недавнего прошлого. Более формально, для заданной входной последовательности x_t схема подразумевает выбор последовательностей из интервала $[t - k - l, t - k]$, где k – размер окна для простого скользящего среднего, а l – гиперпараметр, определяющий размер этого интервала. Результаты применения такой схемы отбора будут приведены в соответствующем разделе.

Заметим отдельно, что перед применением схемы отбора осуществляется предобработка данных, заключающаяся в удалении очевидных выбросов, например, по порогу α (удалим имеющие метку «up» или «down» последовательности, для которых $|R_t| > \alpha$).

3.3 Кодировщик

Кодировщик используется для получения внутренних представлений последовательностей. Он состоит из двух частей.

Для начала зафиксируем входную последовательность в момент времени t : $x_t^{(input)}$ и соответствующие ей выбранные $x_t^{(up)}$, $x_t^{(down)}$, $x_t^{(static)}$. Затем преобразуем их независимо в $h_t^{(input)}$, $h_t^{(up)}$, $h_t^{(down)}$ и $h_t^{(static)}$ с помощью LSTMNet:

$$h_t = LSTMNet(x_t). \quad (3)$$

Здесь верхние индексы не указаны для краткости, а LSTMNet – это LSTM-нейросеть – рекуррентная нейросеть, составленная из LSTM-блоков, внутреннее устройство которых будет описано в подразделе ниже.

Следующим шагом необходимо установить отношения между $h_t^{(input)}$, $h_t^{(up)}$, $h_t^{(down)}$ и $h_t^{(static)}$, за что отвечает вторая часть кодировщика – двунаправленная LSTM-нейросеть. Она пропускает последовательности в прямом ($h_t^{(input)}$, $h_t^{(up)}$, $h_t^{(down)}$, $h_t^{(static)}$) и обратном ($h_t^{(static)}$, $h_t^{(down)}$, $h_t^{(up)}$, $h_t^{(input)}$) порядке, генерируя выходное значение как результат сложения:

$$o_{ti} = h_{ti}^{\rightarrow} + h_{ti}^{\leftarrow} + h_{ti}, \quad (4)$$

где h_{ti}^{\rightarrow} (h_{ti}^{\leftarrow}) – i -ый выход из двунаправленной LSTM-нейросети при прямом(обратном) прохождении, o_{ti} – i -ый выход кодировщика.

3.4 Архитектура LSTM-блока

Традиционные нейронные сети не поддерживают концепцию памяти, и в этом их главный недостаток, проявляющийся при анализе динамических данных, в частности, временных рядов. Решить проблему помогают рекуррентные нейронные сети (Recurrent Neural Networks, RNN), содержащие обратные связи и позволяющие сохранять информацию. Рекуррентную сеть можно рассматривать как несколько копий одной и той же сети, каждая из которых передает информацию последующей копии. Однако по мере роста расстояния между копиями, RNN теряют способность связывать информацию. Резюмируя, такие нейросети хорошо работают только на последовательностях небольшой длины.

Долгая краткосрочная память (LSTM) – особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям. В ее блоке, изображенном на рисунке 3, используется четыре взаимодействующих слоя. Разберем функционирование каждого из них подробнее.

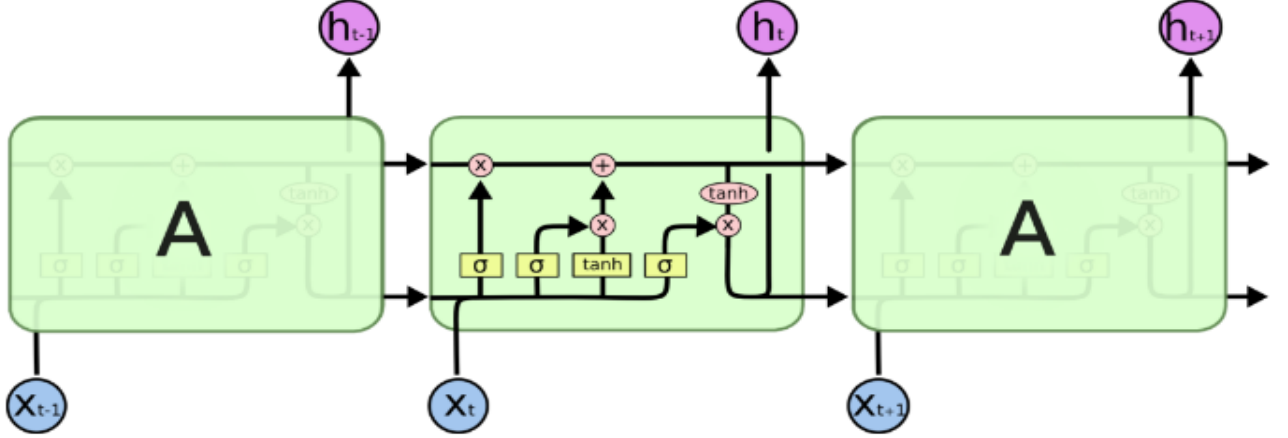


Рис. 3: Внутреннее устройство нейрона LSTM-нейросети

Первый шаг в LSTM – определить, какую информацию можно выбросить из состояния блока. Это решение принимает сигмоидальный слой, называемый «слоем фильтра забывания» (forget gate layer):

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f). \quad (5)$$

Следующий шаг – решить, какая новая информация будет храниться в состоянии блока. Сигмоидальный слой под названием «слой входного фильтра» (input layer gate) определяет, какие значения следует обновить:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (6)$$

На третьем шаге tanh-слой строит вектор новых значений-кандидатов на добавление в состояние блока, затем происходит непосредственно обновление:

$$C_t = f_t C_{t-1} + i_t \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (7)$$

Наконец, нужно решить, какую информацию мы хотим получать на выходе. Выходные

данные основаны на состоянии блока, к ним применяются некоторые фильтры. Сначала сигмоидальный слой решает, какую информацию из состояния блока следует выводить. Затем значения состояния блока проходят через \tanh -слой (приводятся к диапазону от -1 до 1) и перемножаются с выходными значениями сигмоидального слоя, что позволяет выводить только требуемую информацию:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (8)$$

$$h_t = o_t \tanh(C_t). \quad (9)$$

Полученные таким образом h_t и C_t передаются далее по цепочке.

Существует множество модификаций LSTM-нейросетей, однако основной принцип работы сохраняется неизменным.

4 Набор данных

Наборы данных представляют собой высокочастотные (минутные) временные ряды рынка криптовалют (OHLC-цены, объем транзакций, рыночная капитализация), полученные через открытый API.

В ходе проведения экспериментов использовались криптовалюты биткоин, лайткоин и эфириум в долларах США и китайских юанях (BTC/LTC/ETH, USD/CNY).

На рисунке 4 изображены цены закрытия и динамика объема транзакций для криптовалюты BTCUSD.

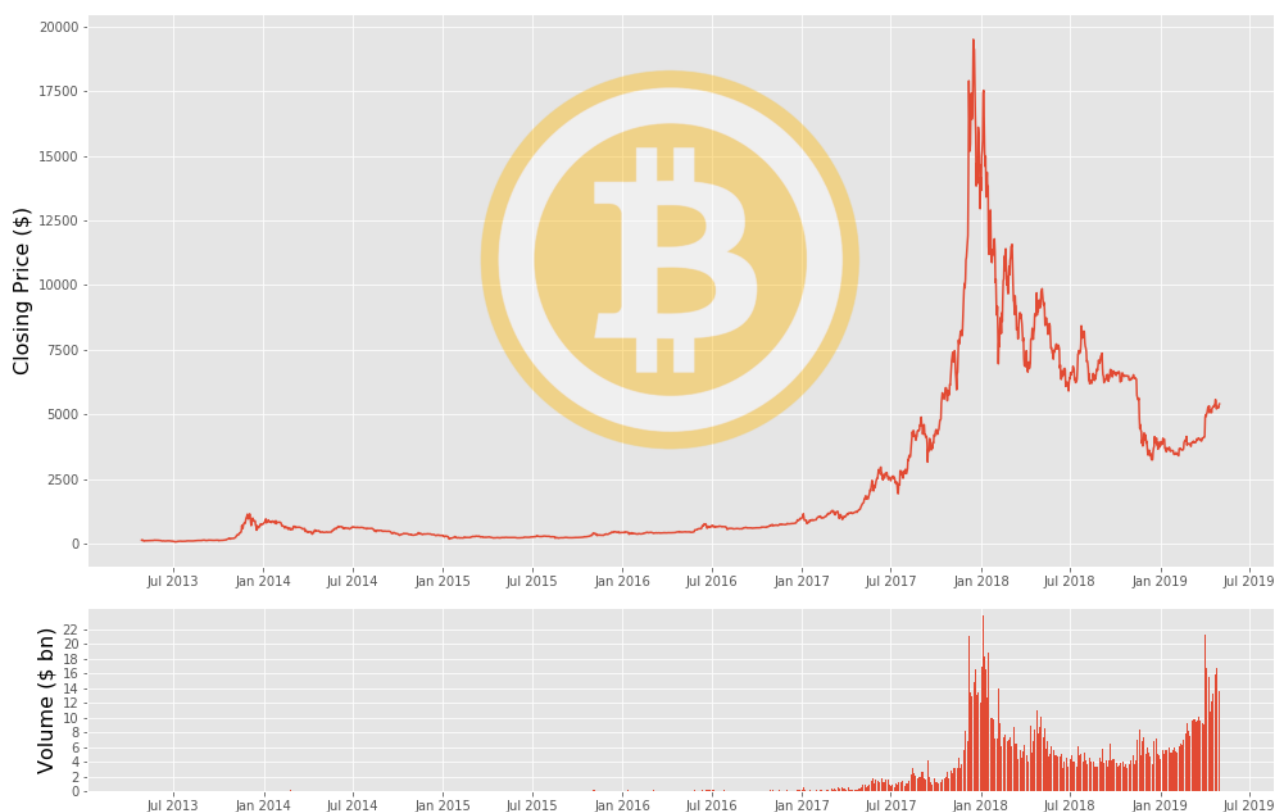


Рис. 4: Цены закрытия и объем транзакций (агрегированные). По горизонтальной оси отмечено время (с 24 апреля 2013 по 1 мая 2019), по вертикальной – цена и объем. Пик приходится на конец 2017, начало изменений в динамике на конец 2016.

5 Предобработка данных

5.1 Входные данные

На вход модели подаются OHLC-временные ряды (Open, High, Low, Close). Как известно, дисбаланс между значениями признаков может вызвать неустойчивость работы модели, ухудшить результаты обучения и замедлить его процесс, поэтому необходимо производить нормализацию данных. Так как изучаемые ряды нестационарны, будем сначала извлекать последовательность из набора, а затем применять к ней нормализацию. В рассматриваемой задаче наиболее оптимальным выбором является минимаксная стратегия нормализации, так как большинство криптовалютных временных рядов имеет скачок в конце 2017 года, и, следовательно, максимум ряда известен:

$$x_{ti}^{norm} = \frac{x_{ti} - \min(x_t)}{\max(x_t) - \min(x_t)}, \quad (10)$$

где индекс i соответствует номеру признака (Open, High, Low или Close), индекс t – временной отметке во входной последовательности, а операции взятия минимума и максимума применяются к компонентам x_{ti} для всех $i = 1, \dots, 4$ на отрезке $[t - j + 1, \dots, t]$, $j = 32$ – длина входной последовательности. В результате получим значения от 0 до 1.

5.2 Выходные данные

Во время обучения модели производилась минимизация кросс-энтропийной функции потерь (стандартный выбор для задачи классификации). Выход – one-hot-векторы, в которых на месте верного класса установлена 1, а все остальные разряды обращены в 0.

Обозначим через m_t среднее цен скользящего окна размера $T = 30$. Тогда целевые метки представимы следующим образом:

$$y_t = \begin{cases} -1, & \text{если } m_t > m_{t+T} + \epsilon \\ 1, & \text{если } m_t < m_{t+T} - \epsilon \\ 0, & \text{иначе} \end{cases}$$

Здесь ϵ – пороговый параметр, вводимый с целью контроля баланса классов (был установлен равным 0.24).

6 Применение моделей

6.1 Настройка гиперпараметров

В качестве функции активации на всех слоях, за исключением выходного, где корректный вывод производит функция Softmax, была использована выпрямленная линейная единица (Rectified Linear Unit (ReLU)), хорошо проявляющая себя на разреженных данных и призванная стабилизировать обучение, в том числе, в глубоких архитектурах. Метод адаптивной инерции (Adam) с шагом обучения $lr = 10^{-3}$ был выбран оптимизатором модели.

Архитектура кодировщика представлена 2-мя слоями по 32 LSTM-блока для каждой из составляющих его частей (всего 4 слоя по 32 нейрона). Базовые архитектуры представлены многослойным перцептроном (MLP) и LSTM-нейросетью, состоящими из 2-х внутренних слоев по 32 нейрона.

Набор данных был разделен на тренировочную (80%), валидационную (10%) и тестовую (10%) выборки. Размер окна l в схеме отбора составил 0.01 часть тренировочной выборки. Для каждого метода было выполнено множественное оценивание, которое производилось на валидационной выборке с использованием техники ранней остановки (Early Stopping). Коэффициент отклонения валидационных результатов между разными запусками сохранялся на уровне 0.1%, что означает устойчивость первых двух значимых знаков на валидационной фазе. Полученная в результате модель подверглась затем тестированию на соответствующей выборке при задействовании стандартного набора метрик (accuracy, точность, полнота, F-мера).

6.2 Результаты тестирования

Значения метрик из стандартного набора, полученные после вычисления вероятностных распределений и выбора классов с наибольшей вероятностью, приведены в таблице ниже.

	Accuracy	Recall	Precision	F1 Score
MLP	0.5594	0.4980	0.5013	0.4821
LSTM	0.5794	0.5499	0.5752	0.5069
RSM	0.6299	0.5573	0.5523	0.5402

Таблица 1: Результаты оценивания моделей для BTCUSD

Как можно видеть из таблицы, максимальное значение F-меры дает RSM-модель. Дан-

Посмотрим на матрицы неточностей (confusion matrix), столбцы которых резервируют-
ся за экспертными решениями, а строки – за решениями классификатора.



Обратим внимание на то, что RSM-модель гораздо меньше смещена к классу «static».

Изучим, какое влияние на эффективность оказывает изменение схемы отбора. Оценим представляемую RSM-модель в двух ее вариациях: с использованием описанного выше (назовем его базовым) и альтернативного методов выбора последовательностей, где альтернативный метод заключается в следующем: будем классифицировать входные последовательности с помощью эталонной тройки из самого начала набора данных. Заметим, что если схема отбора не изменяет показателей эффективности, то это может означать, что динамика торговли не менялась на выбранных экземплярах.

Интересно заметить, что в результате проведения эксперимента показатели работы модели на временных рядах, содержащих OHLC-цены в долларах, практически не изменились, зато на временных рядах с данными в китайских юанях – ухудшились очень сильно.

Подход	Accuracy	Recall	Precision	F1 Score
базовый	0.5399	0.5269	0.5539	0.5159
альтернативный	0.4066	0.4895	0.6111	0.4187

Таблица 2: Оценка моделей на BTCCNY

Исследование этого вопроса показало, что цена BTCCNY в конце 2013 года резко выросла и сильно колебалась в течение нескольких месяцев. Данное сильное колебание, не наблюдающееся во временных рядах BTCUSD, предположительно и послужило причиной ухудшения показателей. Базовая схема отбора имеет преимущество в подобных случаях, так как использует эталонные последовательности из всего прошлого, а не только его из фиксированного промежутка.

6.4 Проверка универсальности

Рассмотрим вопрос об универсальности модели. Будем применять предобученную модель с фиксированными настройками и оценивать ее на новом наборе данных.

Возможно, существовала некоторая взаимосвязь между распределениями одного и того же актива на тренировочной и тестовой выборках. Поэтому имеет смысл посмотреть на результаты применения модели к другим активам.

	Accuracy	Recall	Precision	F1 Score
MLP	0.4952	0.4962	0.5087	0.4940
LSTM	0.5277	0.5367	0.5787	0.5326
RSM	0.5561	0.5539	0.5672	0.5534

Таблица 3: Результаты оценивания моделей на LTCUSD

Воспользуемся тестовым набором данных (того же размера, что и ранее), сформированным для криптовалюты лайткоин (LTCUSD). Модели были обучены на BTCUSD (с тем же размером тренировочной выборки, что и ранее) и не имели во время обучения доступа к данным LTCUSD. В итоге снова приходим к тому, что и LSTM, и RSM выдают приемлемые результаты, при этом RSM является более эффективным по большинству метрик, исключая только метрику точности по причинам, разобранным выше.

7 Заключение

Основной целью данной дипломной работы являлось исследование нового метода обучения – метода случайного отбора – на задаче классификации (предсказания движения цены) временных рядов рынка криптовалют. В итоге было показано, что он имеет высокую эффективность в тех областях применения, где верно предположение о нестационарном характере ряда. Заметим, что данное условие может быть выполнено и в случае финансовых временных рядов общего вида. Был произведен ряд экспериментов с моделями на высокочастотных данных, позволивший детально изучить результативность метода случайного отбора и выявить его превосходство над общеизвестными алгоритмами глубокого обучения, применяющимися к рассматриваемой задаче: MLP и LSTM. Все полученные в ходе экспериментов показатели модели могут быть улучшены в дальнейшем путем ее более тонкого настраивания или расширения входного пространства за счет вовлечения в него новых признаков, таких, например, как данные самого блокчейна (средний размер блока, количество пользовательских адресов, количество транзакций, доход майнеров), данные анализа тональности (интерес со временем слова «биткойн»), данные фондовых индексов (Доу Джонса, S&P 500 и др.).

Список литературы

- [1] *Ефремова Е.А., Дунаев Е.В.* Применение нейронных сетей для прогнозирования финансовых временных рядов. // Доклады ТУСУР. – С.192-196. – 2004.
- [2] *Лоскутов А.Ю., Козлов А.А., Хаханов Ю.М.* Энтропия и прогноз временных рядов в теории динамических систем. // Известия Вузов. ПНД. – С.98-111. – 2009.
- [3] *Bariviera, Aurelio F., Maria Jose Basgall, Waldo Hasperue, and Marcelo Naiouf.* Some stylized facts of the bitcoin market. // Physica A: Statistical Mechanics and its Applications 484: 82–90. – 2017.
- [4] *Bengio Y.* Neural Networks for Speech and Sequence Recognition. // Thomson, London. – 1995.
- [5] *Dixon, Matthew, Diego Klabjan, and Jin Hoon Bang.* Classification-based financial markets prediction using deep neural networks. // Algorithmic Finance 6: 67–77. – 2017.
- [6] *Gao, P., Woo, W., Dlay, S.* Non-linear independent component analysis using series reversion and weierstrass network. // IEE Proceedings-Vision, Image and Signal Processing 153(2), 115–131. – 2006.
- [7] *Gers, Felix A., Jürgen Schmidhuber, and Fred A. Cummins.* Learning to forget: Continual prediction with lstm. // Neural Computation 12: 2451–2471. – 2000.
- [8] *Gkillas, Konstantinos, and Paraskevi Katsiampa.* An application of extreme value theory to cryptocurrencies. // Economics Letters 164: 109–111. – 2018.
- [9] *Greff, Klaus, Rupesh K. Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber.* Lstm: A search space odyssey. // IEEE Transactions on Neural Networks and Learning Systems 28: 2222–2232. – 2017.
- [10] *Hilliard, Nathan, Lawrence Phillips, Scott Howland, Artem Yankov, Courtney D. Corley, and Nathan O. Hodas.* Few-shot learning with metric-agnostic conditional embeddings. // 2018.
- [11] *Hochreiter S. and Schmidhuber J.* Long short-term memory. // Neural computation, 9(8):1735–1780. – 1997.

- [12] *Kristoufek, Ladislav*. On bitcoin markets (in)efficiency and its evolution. // *Physica A: Statistical Mechanics and its Applications* 503: 257–262. – 2018.
- [13] *Lake, Brenden M., Ruslan Salakhutdinov, and Joshua B. Tenenbaum*. Human-level concept learning through probabilistic program induction. *Science* 350: 1332–1338. – 2015.
- [14] *Li, Fe-Fei, Rob Fergus, and Pietro Perona*. A bayesian approach to unsupervised one-shot learning of object categories. // Paper presented at the 2003 Ninth IEEE International Conference on Computer Vision, Nice, France, October 13–16, pp. 1134–1141.
- [15] *Mäkinen, Milla, Juho Kanninen, Moncef Gabbouj, and Alexandros Iosifidis*. Forecasting of jump arrivals in stock prices: New attention-based network architecture using limit order book data. // 2018.
- [16] *Schuster, Mike, and Kuldip K Paliwal*. Bidirectional recurrent neural networks. // *IEEE Transactions on Signal Processing* 45: 2673–2681. – 1997.
- [17] *Sirignano, Justin, and Rama Cont*. Universal Features of Price Formation in Financial Markets: Perspectives From Deep Learning. // 2018.
- [18] *Zhang, Zihao, Stefan Zohren, and Stephen Roberts*. Deeplob: Deep convolutional neural networks for limit order books. // 2018.