Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

**Факультет экономических наук**

**КУРСОВАЯ РАБОТА**

**Зависимость между временным окном технических индикаторов и горизонтом предсказания модели в модели глубокого обучения LSTM на российском рынке**

образовательная программа «Экономика»

Выполнила:

Андреичева Полина Вячеславовна

Руководитель:

преподаватель факультета

экономических наук и

базовой кафедры инфраструктуры

финансовых рынков

Галенская Кристина Владимировна

Москва, 2021

Оглавление

[Введение 3](#_Toc75211628)

[Обзор литературы 4](#_Toc75211629)

[Набор данных 8](#_Toc75211630)

[Методология 15](#_Toc75211631)

[Технические индикаторы 15](#_Toc75211632)

[Нейронные сети 24](#_Toc75211633)

[RNN 26](#_Toc75211634)

[LSTM 29](#_Toc75211635)

[Выбор фреймворка 32](#_Toc75211636)

[Конфигурация модели 33](#_Toc75211637)

[Метрика качества 35](#_Toc75211638)

[Классификация 35](#_Toc75211639)

[Горизонт предсказания 35](#_Toc75211640)

[Результаты 36](#_Toc75211641)

[Список литературы 38](#_Toc75211642)

# Введение

Модели глубокого обучения широко изучались и показывали высокую прибыльность при тестировании. Тем не менее их работа слабо понятна, и может быть изучена на основе большого количества экспериментов, проверяющих действие на качество модели разных переменных и параметров модели.

Целью данной работы является изучение зависимости временного окна индикатора и горизонта предсказания модели. Горизонт предсказание — это количество наблюдений, которые модель предсказывает. Временное окно – это промежуток времени, за который индикатор был построен. В предыдущих работах лучшие результаты получали те модели, в которых горизонт предсказания был равен временному окну индикатора. Однако в них были существенные отличия от данной работы. Во-первых, большинство из них были основаны на моделях машинного обучения, а не глубокого обучения. Во-вторых, в основном брались дневные данных, то есть поведение на коротких временных промежутках может существенно отличаться. Наконец, хорошо исследован американский рынок по данной теме, а на российском рынке данная зависимость пока не изучена. Таким образом, исследование поможет закрыть это пробелы в литературе.

Ожидается, что результаты совпадут с предыдущими исследованиями по теме, и в лучших моделям горизонт предсказания будет равен временному окну индикатора. В дальнейшем исследование этой зависимости может быть использована для построения качественной и прибыльной модели. Исследование, основанное на большом горизонте предсказания, поможет в построении более прибыльной стратегии на несколько наблюдений вперед.

# Обзор литературы

В статье «Financial time series forecasting with deep learning: A systematic literature review: 2005–2019» (Sezer et al., 2020), проводится первый в научной литературе систематический обзор глубокого обучения для предсказания финансовых временных рядов, в который включены 140 работ. В исследование были включены публикации из научных журналов, докторские и главы из книг. Самыми популярными в литературе моделями стали: RNN (Recurrent Neural Networks) с 51% всех публикаций, Deep Multilayer Perceptron (DMLP) с 20.8%, CNN с 14%. В самой широко изученной категории RNN 60% моделей было на базе архитектуры LSTM (Long Short-Term Memory) , которая лучше всех рекуррентных нейросетей подстроена для изучения временных рядов. Так же статья отмечает наличие в литературе более редких моделей таких как Reinforcement Learning based, Deep belief network, Autoencoders, и Restricted Boltzmann machine, некоторые из которых показывали более высокие результаты чем классические работы. Кроме того, исследование выявило, что модели глубокого обучения в большинстве работ более эффективны чем модели машинного обучения, однако в значительном числе работ их эффективность сравнительно одинаковая. Исследователи отмечают следующий тренд в данной сфере модели глубокого обучения становятся более популярными чем модели машинного обучения. Так же они предсказывают, что Natural Language Processing модели и гибридные семантические модели будут более популярны в будущих работах. В работе показано, что существует множество исследований, покрывающих фондовый, валютный, срочный, товарный рынок, индексы и Forex – все направления не настолько исследованы на российском рынке при помощи моделей глубинного обучения. Практически все работы, рассмотренные в обзоре, изучали рынки США, Японии и Китая, что говорит о недостатке литературы о развивающихся рынках и, в частности, российского рынка, на котором качество моделей и выводы могут отличаться. Тем не менее публикация хорошо систематизирует исследования в данной сфере.

В научной литературе практически не изучена применимость моделей глубокого обучения для развивающихся рынков, какие модели более эффективны, а также возможные отличия между моделями. Однако в последние годы появилось 2 публикации, рассматривающие данную тему. Работа «Predictive intraday correlations in stable and volatile market environments: Evidence from deep learning» (Moews, 2020) исследует предсказательную способность модели глубокого обучения в зависимости от волатильности рынка. В работе используется архитектура автокодировщика для того, чтобы изучить разницу в качестве модели глубокого обучения для предсказания кросс-корреляций американских акций в периоды стабильности и кризисов. Модель была обучена на исторических данных 450 американских компаний из списка S&P 500 за период 4.5 лет до финансового кризиса, а далее протестирована в периоде с июля 2007 до конца 2008. Набор данных включал в себя следующие переменные: идентификационный номер акции, дату и время с точностью до миллисекунды, среднюю цену транзакции, самую высокую цену спроса, самую низкую цену предложения и объем торгов. Исследователи пришли к выводу, что в периоды кризисов лучшее качество показала модель, предсказывающая цену закрытия, чем модель, предсказывающая поминутные цены. Тем не менее, обе модели предоставляют возможности для получения прибыли выше средней на рынке, а именно выше 50%, при помощи модели глубокого обучения, которая исследует взаимосвязи между котировками акций. Из этого можно сделать вывод, что даже на развивающихся и волатильных рынках есть место для моделей глубокого обучения, и они могут быть эффективны для предсказания разных временных интервалов.

Другим исследованием, которое изучает эффективность моделей глубокого обучения на развитых и развивающихся рынках, является статья «A deep learning framework for financial time series using stacked autoencoders and long-short term memory» (Bao, Yue, Rao, 2017). Статья рассматривает 2 популярные в литературе модели: LSTM и RNN. А также две гибридные: WLSTM – комбинация wavelet transforms (WT) и LSTM и WSAEs-LSTM – комбинация WT, LSTM и stacked autoencoders (SAEs). Модели были протестированы на 6 индексах разных стран, которые разделены на развитые рынки, частично развитые рынки и развивающиеся рынки. Переменные представляют собой OHLCV -показатели, 10 технических показателей и 2 макроэкономических показателей за период 8 лет. В целом качество каждой модели варьируется при тестировании на индексах, но не имеет сильной зависимости от развитости рынка. Например, прибыльность моделей при тестировании на индексах S&P 500 Index и Nifty 50 Index практически одинаковая. На основе 6 индексов нельзя сделать выводы о повсеместной применимости алгоритмов глубокого обучения, но на предоставленных данные показано, что их прибыльность соответствует развитым странам. Другим важным результатом стала средняя годовая доходность разных моделей. Самая высокая доходность была у разработанной авторами модели WSAEs-LSTM – 45-64%, WLSTM – 23-39%, LSTM – 11-25%, RNN – 7-16%, Buy-and-hold показывает результат от -1% до 6%. Эти результаты говорят о том, что более сложные модели могут приносить большую доходность, а также модели машинного обучения могут приносить доходность превышающую стратегию buy-and-hold в десятки раз.

Один из немногих примеров опубликованной модели глубокого обучения на российском рынке индексов или акций представлена в работа «Using Reinforcement Learning in the Algorithmic Trading Problem» (Ponomarev et al., 2020). В статье изучена усовершенствованная модель LSTM, после которой применяется недавно разработанный алгоритм «asynchronous advantage actor-critic (A3C)» (Mnih et al., 2016). Данная статья так же стала первым применением A3C для алгоритмической торговли. Несколько моделей были обучены на полном стакане заявок за полгода с 15 сентября по 15 декабря 2015 года на фьючерс индекса RTS, далее тестирована в течение следующих 6 месяцев. Полный стакан заявок включает в себя не только информацию о совершенных сделках, но и о всех открытых заявках. Об открытых заявках и сделках по ним есть следующая информация: их количество, цена и время подачи/совершения сделки. Получение таких данных осложнено тем, что они либо продаются биржей за высокую цену, либо собираются каждый торговый день через терминал. Это ограничивает использование некоторых новых моделей с агентами. У лучшей модели прибыль за это время составляет 66% с учетом комиссии за полгода. Данный результат показывает, что есть перспективы для исследования российского рынка при помощи моделей глубокого обучения, и что они могут успешно применяться на российском рынке. Кроме того, работа иллюстрирует необходимость в исследовании базовых моделей на российском рынке, так как многие современные исследования являются усовершенствованием основных моделей.

В научной литературе существуют исследования о предсказательной способности моделей машинного обучения на разных временных интервалах. В то же время на данный момент нет цитируемых исследований, изучающий этот эффект в моделях глубокого обучения. В исследовании «Forecasting price movements using technical indicators: Investigating the impact of varying input window length» (Shynkevich et al., 2020) изучена связь временного промежутка, за который устанавливается технический индикатор такой как 10-дневная скользящая средняя, и горизонта прогнозирования. Набор данных состоял из цен закрытия 50 акций за 10 лет. Данные содержат OHLCV-показатели за каждый день в течение этого периода с поправкой на дивиденды. При тестировании эти данные были поделены на промежутки от 3 до 30 дней, а также использованы 3 модели машинного обучения (SVM, ANN, kNN), качество каждой из них измерено с помощью 4 широко используемых метрик. Результатом исследования стало нахождение следующей зависимости: лучшее качество достигается при совпадении временного промежутка, за который устанавливается технический, и горизонта прогнозирования. Существуют другие менее цитируемые исследования, которые подтверждают данный вывод. Кроме того, в одной так же мало цитируемой работе, исследующей качество модели глубокого обучения на разных временных промежутках, не было найдено зависимости между увеличением рассматриваемого промежутка и предсказательной способности. Следовательно, нужно учитывать и исследовать в дальнейшей работе предпосылки о возможной нелинейной связи данных показателей.

Таким образом, многие исследования подтверждают, что модели глубокого обучения показывают высокую предсказательную способность на развитых рынках. Мало изучена применимость данных алгоритмов на развивающихся рынках, а на российском практически не изучена, тем не менее существующие исследования указывают на примерно одинаковую точность прогноза. В некоторых исследованиях показано, что имеют значения интервалы прогнозов, горизонт прогноза и период обучения модели. В сравнении с исследованиями крупнейших мировых рынков на данный момент не хватает исследования базовых алгоритмов на российских фондовых, срочных и валютных рынках, а также сравнения их эффективности. Однако, стоит отметить, что наибольшую цитируемость набирают публикации с усовершенствованными алгоритмами. При выборе методологии можно ориентироваться как на базовые исследования крупнейших рынков, так и комбинировать разные модели для достижения лучшего качества. Большинство исследований совершенствуют и комбинируют именно модель LSTM, что подчеркивает необходимость изучения применимости ее на российском рынке. В литературе объединялись практически все возможные модели глубокого обучения, а также комбинировались данные алгоритмы и статистические методы. Кроме того, множество исследований в этой сфере опираются на новшества в области компьютерных наук. Из первой работы, которая систематизирует текущую литературу, известно, что примерно половина работ использует только OHLCV-показатели, остальные в подавляющем большинстве включают только технические индикаторы и небольшая часть добавляет несколько макроэкономические показатели. Возможно, в ресурсно-ориентированной экономике включение цен на ресурсы и некоторые валюты улучшит качество, это пока что было исследовано только отчетами на конференциях, а не опубликованными в журналах работами.

# Набор данных

Для исследования выбраны 20 самых ликвидных акций на российском рынке. Период исследования составляет 5 месяцев с августа по декабрь 2020 года включительно. Собраны поминутные значения следующих показателей: цены открытия, цены закрытия, минимальное и максимальное значения и объем торгов. Всего по каждой акции более 80 тысяч наблюдений. В данных не было пустых значений. Однако при составлении индикаторов получались ошибки из-за деления на ноль. Такие значения составляли менее 2% от всех наблюдений, поэтому были интерполированы, то есть для заполнения значения было взято среднее по соседним значениям. Также все значения были нормированы, как и в других исследованиях. Другими словами, их медиана была приравнена к 0, а дисперсия к 1.

Набор данных состоит из 20 компаний с наибольшей рыночной капитализацией на российском рынке:

1. Сбербанк России (SBER)
2. Газпром (GAZP)
3. Лукойл (LKOH)
4. ГМК Норильский никель (GMKN)
5. ЯНДЕКС Н.В. (YNDX)
6. Роснефть (ROSN)
7. Татнефть (TATN)
8. Магнит (MGNT)
9. Сургутнефтегаз (SNGS)
10. НОВАТЭК (NVTK)
11. Polymetal International (POLY)
12. Аэрофлот (AFLT)
13. Московская Биржа (IMOEX)
14. АЛРОСА (ALRS)
15. ВТБ (VTBR)
16. ИНТЕР РАО (IRAO)
17. МТС (MTSS)
18. Северсталь (CHMF)
19. НЛМК (NLMK)
20. АФК Система (AFKS)

Котировки данных акций визуализированы на Рисунке 1. Собраны поминутные значения цены открытия, максимальной цены, минимальной цены, цены закрытия и количество сделок, так же называемые OHLCV-показателями, за период с 01.08.2020 по 31.12.2020 включительно. Данный период характеризуется волатильностью на рынке, так как в это время на рынке происходили корректировки после падения летом 2020 года. У всех акций резко выделяется скачок в ноябре 2020 года, такие резкие колебания трудны для предсказания, так как встречаются редко, и модели не на чем обучаться. Тем не менее такие моменты очень важны, так как информативны для понимания паттернов. Дисперсия разная у каждой акции, но визуально не заметно ее изменение со временем, то есть модель не должна сильно меняться со временем.

В целом почти все выбранные цены акций имеют повышающийся или нисходящий тренд за данный период, однако есть достаточно много периодов, когда цена колеблется между двумя значениями без явно выраженного тренда. Данное явление так же называется движение внутри ценового канала. Индикаторы тренда имеют хорошую предсказательную способность, когда не происходит такого статичного движения. В то же время осцилляторы хорошо работают, когда тренд не заметен. В данном случае на таких неоднородных данных важно учесть оба типа технических индикаторов.

Изображение выглядит как диван, другой

Автоматически созданное описание

Рисунок 1. Котировки 20 выбранных акций за период с августа по декабрь 2020 года. Источник: Инвестиционная компания «ФИНАМ» [Электронный ресурс] //: https://www.finam.ru/profile/moex-akcii/alrosa-ao/export/ (дата обращения: 24.02.2021).

Данные в таком формате тяжелы для обучения и предсказания модели, так как они высоко неоднородны, и модель часто встречает значения, которые раньше не встречала. По этой причине лучше всего при предсказании работать с доходностью, то есть с первой разностью ряда. Ниже представлена доходность всех выбранных акций.

Изображение выглядит как окно, электроника, дисплей

Автоматически созданное описание

Рисунок 2. Первая разность всех выбранных акций за период с августа по декабрь 2020 года. Источник: расчеты автора.

Другим важным показателем являются объемы торгов, изображенные на Рисунке 3, данный показатель так же имел сильный скачек в ноябре 2020 у всех акций.

Изображение выглядит как окно, дисплей

Автоматически созданное описание

Рисунок 3. Объемы торгов всех выбранных акций за период с августа по декабрь 2020 года. Источник: расчеты автора.

Финансовые данные часто обладают годовой и месячной сезонностью, так же могут быть недельные, дневные и часовые паттерны. На Рисунке 4 изображены диаграммы размаха доходности акций по дням недели, начиная с понедельника. Доходности некоторых акций двигаются одинаково: доходность AFLT, IMOEX, IRAO, POLY постепенно понижается в течение недели. Однако, у большинства акция отсутствует выраженная разница между днями недели. В целом цены акции не имеют определенного тренда внутри недели, поэтому нет смысла включать в модель фиктивные переменные с днями недели.

Изображение выглядит как окно

Автоматически созданное описание

Рисунок 4. Доходность всех акций по дням недели. Источник: расчеты автора.

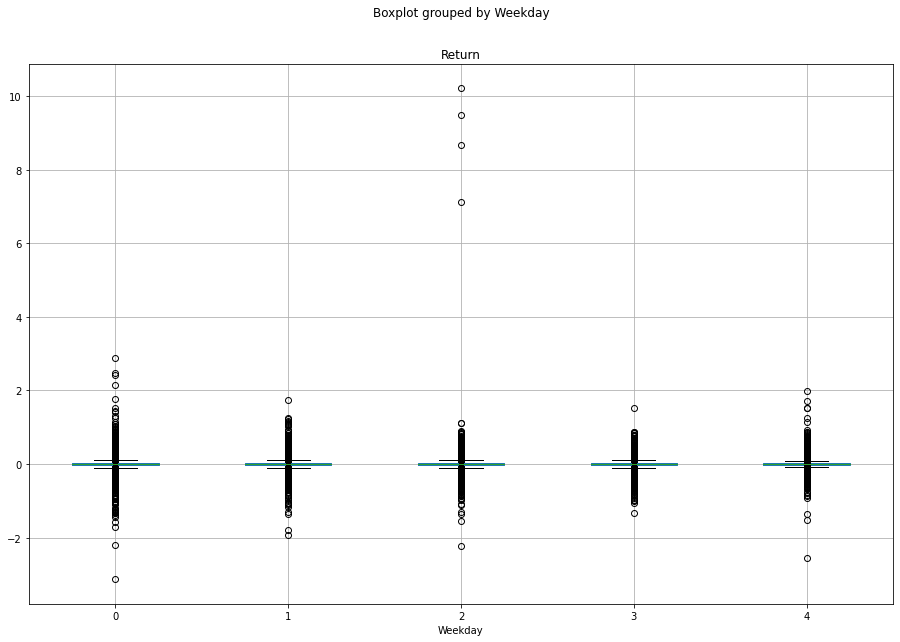


Рисунок 5. Доходность всех акций на одном графике. Источник: расчеты автора.

Доходность во всей выборке в целом на Рисунке 5 подтверждает, что нет значительной разницы между днями недели. В пятницу доходность немного меньше, в среду несколько больших выбросов, но во всем остальном дни схожи.

Кроме того, часто акции имеют внутридневные паттерны. На Рисунке 6 показана дневная сезонность, по которой видно, что серьезные изменения поведения есть только между 11:00 и 12:00, когда цена акции падает, а после 12:00 возвращается на прошлые значения. В первую половину дня цены обычно немного ниже, чем во вторую. Скорее всего это связано с перерывом в торгах. Самая высокая цена обычно около 17:00, а потом немного снижается к концу дня.

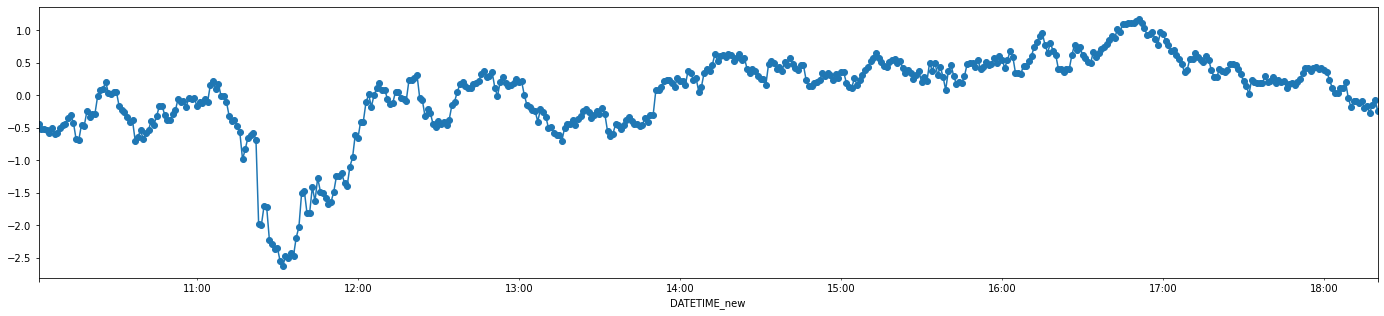


Рисунок 6. Дневная сезонность. Источник: расчеты автора.

Внутри одного часа так же можно обнаружить сезонность, как показано на Рисунке 7. Как правило, с 15 минут после начала каждого часа цена акций повышается, а потом спустя полчаса на сороковой минуте падает. В целом, в часовых и дневных данных обнаружены паттерны, которые, возможно, улучшат качество модели.

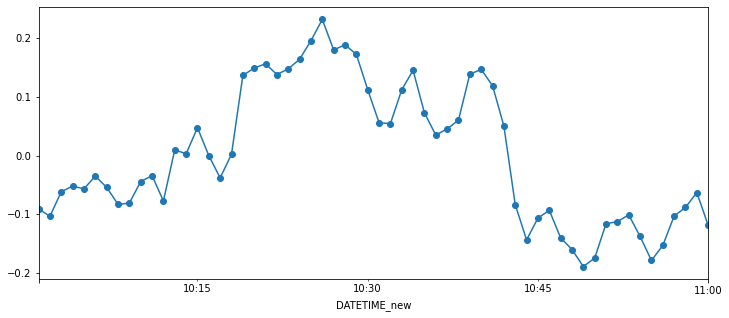


Рисунок 7. Часовая сезонность. Источник: расчеты автора.

# Методология

## Технические индикаторы

Технические индикаторы строятся на основе прошлых цен и объема торгов для предсказания цен, имеется множество теоретических и эмпирических подтверждений их предсказательной способности в литературе. Теоретическое объяснение этому сводится, как правило, к неэффективности финансового рынка. Однако, предложено несколько моделей, по-разному объясняющих данную взаимосвязь. Первый вид модели предполагает, что при помощи технических индикаторов можно извлечь непубличную информацию о фундаментальной стоимости ценной бумаги (Treynor, 1985). Второй вид модели не утверждает наличие связи между техническими индикаторами и фундаментальной стоимостью, но говорит о том, что технические индикаторы формируют ожидания инвесторов и, как следствие, спекулятивное поведение, не связанное с фундаментальным анализом (Giovanni, 2011). Третий вид модели заключается в том, что сделки, отражающие положительные ожидания инвесторов, могут вызывать чрезмерные или, наоборот, недостаточные реакции, что создает паттерны, которые технические индикаторы могут выявить (Hong, 1999, Soros, 2015). Четвертый тип модели относится к настроению инвесторов, исследователи пришли к выводу, что настроения инвесторов могут объяснить премию за риск, а технические индикаторы успешно предсказывают изменения в настроениях инвесторов (Keynes, 1936, Neely, 2014).

Большинство работ, которые предсказывают котировки по LSTM-модели, включают так же технические индикаторы. Причем выбор количества технических индикаторов широко варьируется. Ниже приведены несколько работ, выпущенные за последние 5 лет, где моделью для предсказания являлась LSTM.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Номер работы | Работа | Набор данных и изучаемый рынок | Период | Тип данных | Набор показателей |
| 1 | (Agrawal et al., 2019) | 3 акции на индийском рынке | 2016 | Дневные | OHLCV и 2 ТИ |
| 2 | (Chen et al., 2015) | 3049 акций на китайском рынке | 1990–2015 | Дневные | OHLCV |
| 3 | (Dezsi, 2016) | 1 акция на румынском рынке | 2001–2016 | Дневные | OHLCV |
| 4 | (Borovkova, 2015) | 22 акций крупных компаний на американском рынке | 2017 | Пятиминутные | OHLCV и 12 ТИ |
| 5 | (Chen et al., 2018) | 1 фьючерс на китайский индекс CSI 300 | 2017 | Минутные | OHLCV и 1 ТИ |
| 6 | (Nelson et al., 2017) | 5 акций на бразильском рынке | 2008–2015 | Пятнадцати-минутные | OHLCV и 175 ТИ |
| 7 | (Zhou et al., 2018) | 42 акции на китайском рынке | 2016 | Минутные | OHLCV и 9 ТИ |
| 8 | (Tan et al., 2020) | 1 фьючерс на китайском рынке | 2016 | Получасовые | OHLCV и 21 ТИ |
| 9 | (Khare et al., 2017) | 10 акций на Нью-Йоркской бирже | 1 год | Минутные | Цены закрытия и 8 ТИ |

Таблица 1. Примеры работ по применению моделей глубокого обучения для предсказания финансовых временных рядов.

Как видно, число включенных технических индикаторов не связано с количеством выбранных бумаг или типом данных. В основном же исследователи ограничиваются использованием 12 индикаторами. Довольно много работ ограничиваются только OHLCV-переменными, но, если технические индикаторы все-таки добавляются к набору данных, их число, как правило, составляет от 9 до 12.

Кроме того, существует несколько видов технических индикаторов: индикаторы тренда, индикаторы скорости изменения, осцилляторы. Индикаторы тренда используются для выявления тренда, могут быть неэффективны, когда цена акции на протяжение долгого времени находится внутри канала, то есть колеблется в одном интервале. Во время выбранного периода практически все цены акций имели понижающийся или повышающийся тренд, поэтому данная группа индикаторов скорее всего будет уместна.

Изображение выглядит как диван, другой

Автоматически созданное описание

Рисунок 8. Котировки 20 выбранных акций за период с августа по декабрь 2020 года. Источник: Инвестиционная компания «ФИНАМ» [Электронный ресурс] //: https://www.finam.ru/profile/moex-akcii/alrosa-ao/export/ (дата обращения: 24.02.2021).

Другой тип индикаторов – индикаторы скорости изменения или индикаторы моментума – указывает на скорость изменения цены бумаги. На основе данного индикатора можно судить об изменении тренда. Кроме того, данный тип индикаторов лучше интерпретируется совместно с индикаторами тренда, так их пересечение говорит об изменении тренда, то есть совместное использование данных индикаторов в модели может значительно улучшить ее. Осцилляторы, третий тип индикаторов, показывают на отклонение цены от среднего значения. Они указывают на приближение коррекции или направление фазы колебания цены. Осцилляторы полезны для анализа в периоды отсутствия выраженных трендов. Данные периоды визуально нечасто встречаются выбранном периоде.

Одним из самых часто используемых в литературе набором технических индикаторов является набор из работы «Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange» (Kara, 2011). Из данного наборы удалены индикаторы, в которых отсутствует временное окно, так как они не входят в сферу данного исследования. Кроме того, добавлены технические индикаторы из исследования о связи длинны временного окна технического индикатора и горизонта предсказания (Shynkevich, 2017), а именно Average Directional Movement Index и Exponential Moving Average. В упомянутом выше популярном наборе был один индикатор объема, но у него отсутствовало временное окно, поэтому взят произвольный индикатор с временным окном Force Index.

|  |  |  |  |
| --- | --- | --- | --- |
| Индикаторы тренда | Индикаторы объема | Индикаторы волатильности | Индикаторы моментум |
| Simple Moving Average | Force Index | Average True Range | Relative Strength Index |
| Exponential Moving Average |  |  | The William’s %R oscillator |
| Average Directional Movement Index |  |  | Price rate-of-change |
| Commodity Channel Index |  |  | Stochastic %K |
|  |  |  | Stochastic %D |

Таблица 2. Набор построенных индикаторов и их типы.

Формулы для расчета технических индикаторов в среде Python представлены ниже.

1. Простое скользящее среднее (Simple Moving Average) является среднем значением цены за определенный период:

где это цена закрытия дня t, n – временное окно индикатора.

1. Экспоненциальное скользящее среднее (Exponential Moving Average) отличается от простого скользящего среднего тем, что веса предыдущих цен экспоненциально уменьшаются:

где и n – временное окно индикатора

1. Средний истинный диапазон (Average True Range) отражает волатильность цены:

где , и это максимальное, минимальное значение цены и цена закрытия за минуту t, n – временное окно индикатора.

1. Индекс Среднего Направления Движения (Average Directional Movement Index) помогает определить наличие ценовой тенденции. Он составляется на основе положительно и отрицательно направленных движений индикаторов:

где и представляют собой положительно и негативно направленные движения.

1. Индекс Товарного Канала (Commodity Channel Index) измеряет отклонение цены инструмента от его среднестатистической цены, то есть занижена или завышена ли цена на основе скользящей средней:

где и равно простому скользящему среднему значений за n минут в соответствии с временным окном.

1. Индекс Относительной Силы (Relative Strength Index) это следующий за ценой осциллятор, который колеблется в диапазоне от 0 до 100, определяет продолжит ли цена двигаться по существующему тренду:

где и вычисляются по тому же принципу, что и при расчете Индекса Среднего Направления Движения (ADMI).

1. Процентный диапазон Вильямса (Williams’ Percent Range, Williams’ %R) определять состояния перекупленности или перепроданности акции, а также дающий сигналы о развороте рынка:
2. Price rate-of-change (ROC) измеряет темп изменения цены:
3. Stochastic %K – это индикатор импульса, который показывает переоценена или недооценена ценная бумага:
4. Stochastic %D указывает на разворот рынка:
5. Force Index. Acting as leading indicator of price movements.

Money Flow Multiplier = [(Close - Low) - (High - Close)] /(High - Low)

Money Flow Volume = Money Flow Multiplier x Volume for the Period

ADL = Previous ADL + Current Period's Money Flow Volume

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SMA | EMA | ATR | ADX | CCI | RSI | W%R | ROC | %D | %K | FI |
| Среднее | 25.12 | 25.12 | 0.03 | 52.21 | 0.93 | 50.51 | -47.43 | 0.00 | 0.51 | 0.51 | -192.69 |
| Стандартное отклонение | 4.04 | 4.04 | 0.03 | 18.30 | 83.89 | 23.75 | 35.70 | 0.18 | 0.23 | 0.28 | 27402.64 |
| Минимальное | 19.80 | 19.80 | 0.00 | 2.51 | -4000 | 0.13 | -100 | -2.99 | 0.00 | 0.00 | -1852992 |
| 25% | 21.17 | 21.17 | 0.01 | 38.35 | -84.29 | 32.01 | -80.65 | -0.06 | 0.33 | 0.33 | -204.00 |
| 50% | 25.00 | 25.00 | 0.02 | 50.23 | 3.85 | 51.01 | -45.83 | 0.00 | 0.51 | 0.51 | -0.71 |
| 75% | 28.69 | 28.69 | 0.03 | 64.63 | 86.06 | 68.71 | -12.50 | 0.06 | 0.67 | 0.67 | 156.54 |
| Максимальное | 34.42 | 34.42 | 1.65 | 100.00 | 400.00 | 99.96 | -0.00 | 4.90 | 1.00 | 1.00 | 2949088 |

Таблица 3. Описательные статистики индикаторов для AFKS. Источник: расчеты автора.

Так как не высокочастотных данных значения часто повторяются, может возникнуть много значений NaN или бесконечностей. Nan – это результат вычисления, который не может быть представлен как число. Бесконечность может получиться при делении на ноль. Таблица 4 показывает количество таких значений по всем компаниям для индикаторов с трехминутным временным окном. При больших временных окнах количество таких значений только уменьшается, так как на больших интервалах меньше вероятность повторения значений. Это является серьезной проблемой, так как модели глубокого обучения не способны обучаться, даже если есть одно такое значение. Для устранения этого требуется изучить данные и определиться со способом замены значений NaN и бесконечностей.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SMA | EMA | ATR | ADX | CCI | RSI | W%R | ROC | %D | %K | FI |
| Количество NaN | 0 | 0 | 0 | 172591 | 2216 | 0 | 6314 | 0 | 302997 | 227173 | 0 |
| Количество бесконечностей | 0 | 0 | 0 | 0 | 81300 | 0 | 9 | 0 | 0 | 0 | 0 |

Таблица 4. Количество NaN и бесконечностей в итоговых данных при трехминутном временной окне. Источник: расчеты автора.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | SMA | EMA | ATR | ADX | CCI | RSI | W%R | ROC | %D | %K | FI |
| Количество NaN | 0 | 0 | 0 | 86540 | 0 | 0 | 348 | 0 | 71089 | 64963 | 0 |
| Количество бесконечностей | 0 | 0 | 0 | 0 | 25590 | 0 | 1 | 0 | 0 | 0 | 0 |

Таблица 5. Количество NaN и бесконечностей в итоговых данных при семиминутном временной окне. Источник: расчеты автора.

Всего в каждом столбце 1655199 значений, таким образом, при трехминутном временном окне количество NaN может достигать 18% от всех значений, как например у технического индикатора Stochastic %D. Тем не менее на больших временных окнах такие значения перестают быть значимыми и любой метод может быть применим к ним. Данная особенность работы с техническими индикаторами на высокочастотных данными значительно осложняет работу. Полностью отказаться от некоторых технических индикаторов может быть хорошим решением проблемы, однако потеряется большое количество информации, так как на этапе тестирования отказ от индикаторов значительно снижает качество модели. Кроме того, исключение данных индикаторов из исследования совсем приведет к большой потере информации при работе с большими временными окнами.

Так как для построения модели глубокого обучения такие значения неприемлемы, разработаны различные способы устранения пропущенных значений. Первый возможный вариант — это удаление всех наблюдений с пропущенными значениями. Модель получается робастной, но может потеряться много информации, особенно если таких значений много. Второй способ – заменить значения средней или медианой, таким образом можно сохранить намного больше информации чем в первом методе, однако она может быть значительно искажена. Третий способ — это генерирование новых значений, он очень хорошо подходит для временных рядов, так как во временных рядах естественно существует взаимоотношение между несколькими ближайшими наблюдениями. LOCF – способ генерирования пропущенных значений, при котором текущее пропущенное значение заменяется на предыдущее. Данный метод хорошо работает на процессах с длинной памятью. Другой вариант генерации пропущенных значений – интерполяция. При линейной интерполяции пропущенное значение заменяется линейной комбинацией последнего валидного значения перед пропущенным и первым валидным значением после пропущенного. Например, если одно пропущенное значение стоит между двумя валидными, то оно заменяется средним.

Каждый из данных методов является уместным, но для выбора необходимо учитывать дескриптивные статистики и поведение переменных. Например, в случае индикаторов Stochastic %D и Stochastic %K пропущенные значения скорее всего вызваны повторением OHLC-показателей на коротких интервалах. Кроме того, данные индикаторы интерпретируются как импульсы цены, а при отсутствии изменений цены акции отсутствует так же и импульс. В то же время средние значения данных показателей близки к нулю, то есть замена пропущенных значений на ноль будет приемлемым способом замены пропущенных значений, сохраняя при этом теоретический смысл. На Рисунке 10 заметно поведение этих осцилляторов, характеризующееся низкими значениями функции в периоды малых изменений цены акции. Другим индикатором с большим количеством пропущенных значений является Average Directional Movement Index. Визуально данный индикатор нечасто меняет значения и выгляди достаточно сглаженным временным рядом. По этой причине для замены пропущенных значений в данном случае может хорошо подойти линейная интерполяция, так как этот метод хорошо работает на временных рядах. Для всех других показателей количество пропущенных значений незначительно, поэтому можно применять любой из допустимых методов, поэтому для них была применены линейная интерполяция.

Данный метод хорошо объясняется геометрически. На Рисунке 9 показан процесс замещения пропущенного значения путем линейной комбинации валидных значений так, что пропущенное значение лежит на линии, проведенной между ними. В данном случае оси представляют собой время и переменная с пропущенными значениями, которые необходимо сгенерировать.

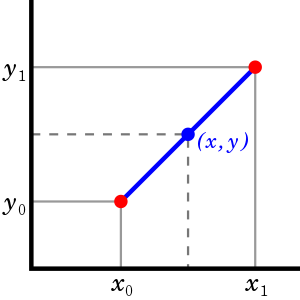


Рисунок 9. Визуализация процесса интерполяции. Источник: Википедия [Электронный ресурс] // Википедия URL: <https://ru.wikipedia.org/wiki/Линейная_интерполяция> (дата обращения: 15.04.21).

Изображение выглядит как здание, окно

Автоматически созданное описание

Рисунок 10. Графики индикаторов за период в 100 минут. Источник: расчеты автора.

## Нейронные сети

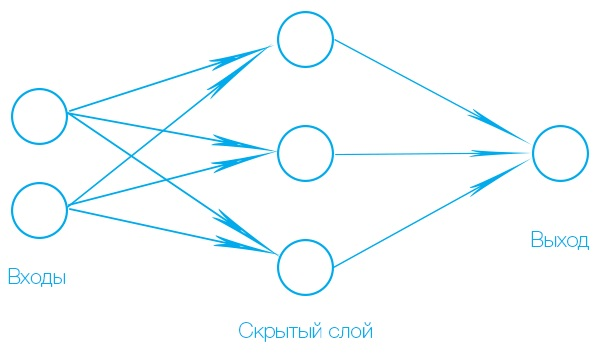


Рисунок 11. Архитектура простой нейронной трехслойной сети. Источник: Modelling and recognition of images [Электронный ресурс] // Простейшие нейронные сети на Python в Visual Studio URL: <https://api-2d3d-cad.com/python-neuron-vs/> (дата обращения: 15.04.21).

Базовое представление нейронной сети с тремя слоями. Например, для задачи анализа финансовых временных рядов Слой 1 (входной слой) состоит из набора независимых OCHLV-переменных. Слой 2 (один скрытый слой) представляет собой атрибуты первого слоя, которые не выражены в нем напрямую. Например, из High и Low модель может извлечь волатильность котировок в данную минуту. Стрелки из входного слоя в скрытый иллюстрируют создание неявных переменных из нескольких явных. Слой 3 (выходной слой) объединяет информацию из скрытых слоев и выдает предсказанное направление движения цены акции. Каждый круг в сети представляет собой нейрон, у которого тип совпадает со слоем, в котором он находится. Другими словами, нейронная сеть состоит из входных, скрытых и выходных нейронов. Обучение нейронной сети представляет собой поиск оптимальных весов, которые объединяют каждый слой нейронов. Основные метода поиска гиперпараметров это метод обратного распространения ошибки, генетический алгоритм и нечеткие алгоритмы.

Глубокое обучение работает следующим образом. Один из основных и самых мощных методов обучение является метод обратного распространения ошибки. На части тренировочных данных, после создания случайных матриц весов считается функция потерь, на основе ошибок формируется градиенты потерь. Наконец, с помощью градиентного спуска устанавливаются новые веса.

Существует две главные категории нейронных сетей: сверточные (CNN) и реккурентные (RNN). Как правило, они применяются для разных задач, но оба типа исследованы в литературе, и их качество сопоставлено. В работе «STOCK PRICE PREDICTION USING LSTM, RNN AND CNN-SLIDING WINDOW MODEL» средний процент ошибки реккурентной нейронной сети составил 5,217%, сверточной – 4,98% (Hiransha, 2017).

В то время в работе сверточных сетей слои соединяются между собой функциями, в рекуррентных сетях функциями так же соединяются прошлые значения переменной.

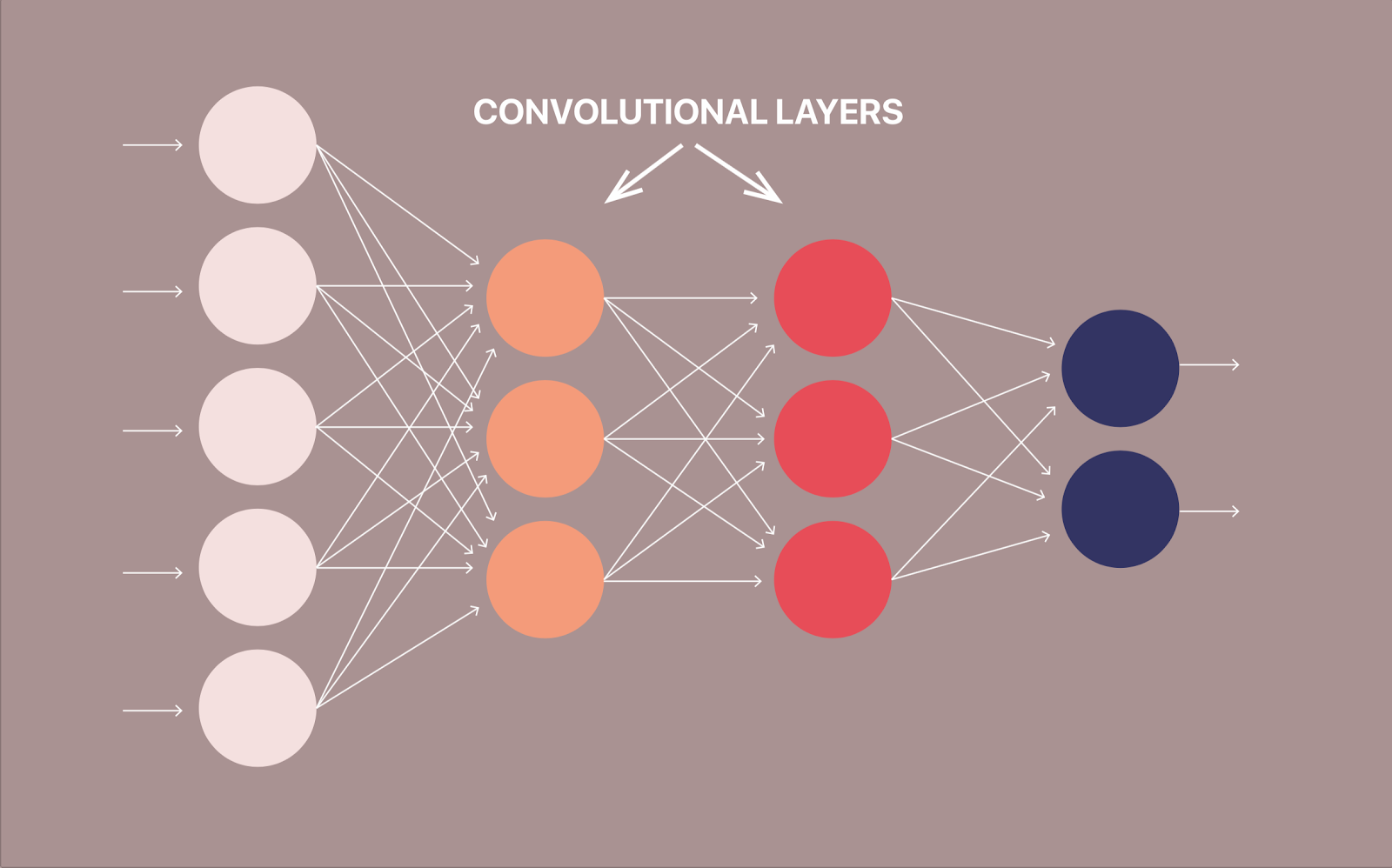


Рисунок 11. Архитектура сверточной нейронный сети. Источник: Lionbridge [Электронный ресурс] // What is the Difference Between CNN and RNN? URL: <https://lionbridge.ai/articles/difference-between-cnn-and-rnn/> (дата обращения: 15.04.21).

CNN отличны от других нейронных сетей наличием сверточных слоев, внутри которых информация трансформируется, после чего передается следующему слою. Трансформация происходит при помощи фильтров, которыми называют матрицы, которые меняют информацию на каждом слое. Данные сети плохо подходят для обработки последовательной информации такой как текст, видео, временные ряды, их применяют для классификации и распознавания объектов и лиц на фотографиях. Во-первых, в них не учитываются предыдущие значения. Во-вторых, они не устроены для изучения информации разной размерности. Тем не менее такой тип нейронных сетей все же используется для предсказания цен акций, но для этой задачи их построение достаточно усложнено. Показатели конвертируются в изображение или создается много сложенных слоев сети.

## RNN

Рекуррентные сети созданы специально для обработки последовательной информации, их особенностью является способность нейронов получать информацию от самих себя предыдущего прохода. Под последовательной информацией понимается следующий набор данных:

где t является временным интервалом в периоде от 1 до (Goodfellow et al., 2016). В данном исследовании последовательностью является период с августа по декабрь 2020 года с интервалом в одну минуту.

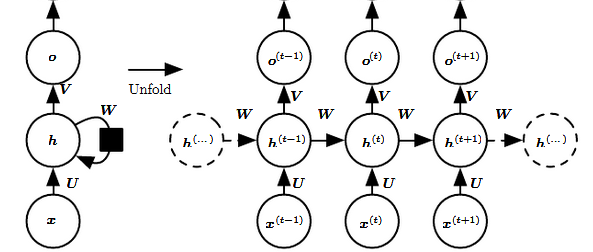


Рисунок 12. Архитектура рекуррентной нейронный сети. Источник: Towards Data Science [Электронный ресурс] // Recurrent Neural Networks (RNNs) URL: <https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85> (дата обращения: 15.04.21).

На Рисунке 12 представлена визуализация рекуррентной сети в сложенном и разложенном виде. Разложенный вид представляет собой сеть из полной последовательности. Источник: Goodfellow I., Bengio Y., Courville A. (2016). Deep learning. MIT Press.

Например, для исследования одного часа, подобная сеть состояла бы из 60 слоев. Входное значение – x(t) – является входным значением нейронной сете на шаге t. В данном случае x(1) является числовым вектором состоящим из OHLCV-показателей и соответствующих технических индикаторов для первой минуты из последовательности. Нейрон из скрытого слоя – h(t) – представляет собой нейрон на шаге t, который выполняет функцию памяти о прошлом состоянии. Рассчитывается такая функция по следующей формуле:

где функция f является функцией активации. Функция активации определяет выходное значение нейрона. По формуле 3 видно, что выходное значение зависит от входного x(t) и от прошлого выходного значения . Так же функция зависит от матриц весов U и W, первая связывает входные и скрытые значения, вторая связывает скрытые значения с прошлыми скрытыми значениями.

Наконец, o(t) на изображении означает выходное значение нейрона. Оно так же связано матрицей с остальными переменными в сети. Матрица весов, соединяющая скрытый слой и выходное значение обозначается V. Эта матрица весов, как и остальные, неизменчива для каждого элемента. Такая связь матрицы и выходного значения представляется в уравнении (4). Изначально матрицы U, W и V генерируется из случайных распределений. Следующим шагом будет передача вперед (Forward pass), которая описывается системой уравнений ниже:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | (2) | |
|  | | (3) | |
|  | | (4) | |
|  | | (5) | |
| Где – нормализованные с помощью операции *softmax* вероятности выходных значений, b и c – смещения, изначально сгенерированные из нулей. В данном примере выбрана функция активации tahn, кроме нее так же могут быть выбраны: сигмоида, ReLu, softmax или другие. Как правило, выбор функции активации происходит при помощи перебора и определения, какая функция дает наибольшее значение. Система уравнений (2)-(5) определяет рекуррентную нейронную сеть с одинаковой длинной входных и выходных значений. Для следующего шага обучения необходимо определить функцию потерь. Функция может быть написано специально под определенную задачу, но одной из универсальных функций потерь является кросс-энтропия, прописанная ниже:   |  |  | | --- | --- | |  | (6) | |  | (7) |   Далее на основе данной функции потерь нужно пересчитать матрицы весов U, W, и V. Для этого, согласно формулам, ниже считаются градиенты функции потерь для данных матриц:   |  |  | | --- | --- | |  | (8) | |  |  | |  | (9) |   Аналогично уравнению (9) считается градиент функций потерь матрицы U. На данном моменте нужно обратить внимание на недостаток RNN для решения задачи предсказания цены акций. На Рисунке 13 представлены производные сигмоидной функции и гиперболического тангенса. Как видно, производные стремятся к нулю при увеличении модуля аргумента. Поэтому с каждым шагом происходит потухание, и не поддерживается тот же темп обучения. Другие функции такие как ReLu наоборот вызывают «взрыв» градиента. Единственным решением данной проблемы является усложнение архитектуры и применения LSTM.  Sigmoid and Tanh Activation Functions · A.I. Dan  Рисунок 13. Графики сигмоидной функции и гиперболического тангенса и их производные. Источник: AI Dan [Электронный ресурс] // Sigmoid and Tanh Activation Functions URL <https://a-i-dan.github.io/math_nn> (дата обращения: 15.04.21).  В теории проблемы с обработкой долговременных зависимостей у RNN быть не должно. Человек может аккуратно подбирать параметры сети для решения искусственные задачи такого типа. К сожалению, на практике обучить RNN этим параметрам кажется невозможным. Эту проблему подробно исследовали Зепп Хохрайтер (Sepp Hochreiter, 1991) и Иошуа Бенджио (Yoshua Bengio) с соавторами (1994); они нашли неоспоримые причины, по которым это может быть сложно. LSTM Long Short-term memory (LSTM) – разновидность архитектуры нейронных сетей, которые отличаются способностью к обучению как долгосрочных так и краткосрочных зависимостей (Goodfellow et al., 2016). | | |  |
| Рисунок 14. Архитектура нейрона LSTM. Источник: Bao, W., Yue, J. and Rao, Y. (2017) ‘A deep learning framework for financial time series using stacked autoencoders and long-short term memory’, PLoS ONE, 12(7), p. 1–24.    Рисунок 15. Повторяющийся модуль в сети RNN. Источник: Bao, W., Yue, J. and Rao, Y. (2017) ‘A deep learning framework for financial time series using stacked autoencoders and long-short term memory’, PLoS ONE, 12(7), p. 1–24.  В сравнении заметно, насколько RNN проще по архитектуре чем LSTM, хотя и схожую цепную структуру. По Рисункам 15 и 16 заметно, что в RNN повторяющийся модуль состоит из одного слоя, который обозначается на картинке прямоугольником, в LSTM уже 4 слоя. На Рисунке 16 кругами обозначаются поточечные операции такие как сложение векторов.    Рисунок 16. Повторяющийся модуль в сети LSTM. – входной вектор, – выходной вектор, – значение скрытого нейрона, – значение входного фильтра, – значение фильтра забывания, - значение выходного фильтра на шаге t. – вектор значений кандидатов на шаге t. Bao, W., Yue, J. and Rao, Y. (2017) ‘A deep learning framework for financial time series using stacked autoencoders and long-short term memory’, PLoS ONE, 12(7), p. 1–24.  Математически Рисунок 16 описывается следующим образом:   1. – входное значение вектора в модуль на шаге t. 2. это матрицы весов. 3. – смещения. 4. – значение скрытого нейрона на шаге t. 5. – значение фильтра забывания на шаге t, рассчитываемое по формуле ниже:  |  |  | | --- | --- | |  | (10) |   Вычисление входного фильтра является первым шагом в работе модуля LSTM. Задачей данного шага является определение, какую информацию сохранить, а от какой избавиться. На данном этапе входное значение шага и скрытое значение нейрона прошлого шага проходят через сигмоидальные слой, называемый «слоем фильтра забывания». В итоге для каждого числа из вектора памяти функция возвращает число от 0 до 1, где 0 соответствует решению полностью сохранить информацию, а 1 – полностью выбросить. Например, в данном случае если произошел разворот тренда, то информация о движении акции до разворота становится неактуальной. Тогда модель скорее всего избавится от большей части этих значений.   1. – значение входного фильтра на шаге t:  |  |  | | --- | --- | |  | (11) |   Следующий шаг позволяет решить, какую информацию хранить в ячейке на основе предыдущего шага. Процесс начинается с определения, какие значения ячейки нужно заменить на новые, при помощи второго сигмоидального слоя, который называется «слоем входного фильтра». Функция так же возвращает значения от 0 до 1, но 0 соответствует решению полностью забыть информацию с прошлого состояния, а 1 – полностью оставить.   1. – вектор значений кандидатов на шаге t:  |  |  | | --- | --- | |  | (12) |   Далее строится вектор значений-кандидатов при помощи tanh функции, которые могут быть добавлены в состояние ячейки. В примере с изменением тренда, желаемым результатом на данном этапе будет создание значений-кандидатов с новыми значениями направления тренда.   1. – вектор памяти на шаге t:  |  |  | | --- | --- | |  | (13) |   После определения, какую информацию стоит сохранить, а также создания новых значений для замены старых, наконец можно изменить старое состояние ячейки на новое . Уравнение (13) объединяет все установленные ранее решения о новом состоянии модели. Старое состояние умножается на слой фильтра забывания , чтобы сохранить только нужную информацию. Новые значения-кандидаты () умножены на слой входного фильтра (). Получается новое состояние ячейки . В данном примере это состояние, когда модель уже изменила направление тренда.   1. – значение выходного фильтра на шаге t:  |  |  | | --- | --- | |  | (14) |   На данном этапе сигмоидальный слой определяет, какая информация будет сохранена на выходе из ячейки.   1. – значение скрытого нейрона на шаге t:  |  |  | | --- | --- | |  | (15) |   Последним шагом вектор памяти проходит через tanh-слой, чтобы вывести значения от -1 до 1. Наконец, этот вектор умножается на значения выходного фильтра на шаге t. При изменении тренда желаемое выходное значение, которое запомнит ячейка памяти. | | |  |
| Выбор фреймворка Среди работ в области предсказания финансовых временных рядов при помощи моделей глубокого обучения 23,7% исследований проводились на фреймворке Keras, что делает его самым популярным выбором (Sezer, 2020). Другими популярными выборами являются PyTorch (22,2%) и Tensorflow (16,3%). Наиболее простыми для написания являются Keras и PyTorch, однако для сложных моделей и создания воспроизводимых экспериментов удобен Tensorflow.   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | Распределенное выполнение | Архитектурные оптимизации | Визуализация | Поддержка сообществом | Портативность | | Tensorflow | ++ | ++ | ++ | ++ | ++ | | Keras | ++ | ++ | + | ++ | ++ | | PyTorch | ++ | ++ | ++ | ++ | ++ | | CNTK | ++ | ++ | + | - | ++ | | MXNet | + | ++ | + | - | ++ | | Torch | - | ++ | + | + | + | | Caffe2 | ++ | ++ | - | - | ++ | | Caffe | - | ++ | + | + | + | | Theano | - | ++ | + | + | + | | | |  |
| Таблица 6. Сравнение фреймворков.  Наиболее подходящим фреймворком для данной задачи является комбинация Keras и PyTorch ввиду того, что модель является стандартной, а данные фреймворки просты в работе. | | |  |
| Конфигурация модели Перед проверкой гипотезы необходимо сконфигурировать модель, чтобы он наилучшим образом подходила для решения поставленной задачи. Кроме того, выбираемые параметры должны соответствовать компутационным возможностям, что так же сделает ее менее долгой и более применимой для работы на высокочастотных данных. В Таблице 7 представлены параметры, а также их описание и роль в модели. | | |  |
| Количество сложенных слоев LSTM, выбор зависит от сложности решаемой задачи. Модель дольше обучается при увеличении количества слоев. | num\_layers | | |
| Вероятность того, что выход любого данного нейрона будет удален. Используется для недопущения переобучения. | dropout | | |
| Количество обучающих примеров за одну итерацию. Чем меньше показатель, тем меньше памяти требуется и меньше времени на обучение, но менее точные оценки. | batch\_size | | |
| Коэффициент скорости обучения, меняется от 0 до 1. При малых значениях модель обучается дольше, слишком большие значения могут привести к неоптимальному решению. | learning\_rate | | |
| Затухание скорости обучения, понижение скорости обучения с итерациями. Нужно, чтобы модель была более стабильной. | learning\_rate\_decay | | |
| Общее количество эпох, качество становится лучше с каждой эпохой, но занимает больше времени. | epoch | | |
| Количество шагов в обучающей итерации. Выбор зависит от свойств данных, выбирается путем перебора. | n\_steps | | |

Таблица 7. Конфигурация модели для всех экспериментов в PyTorch.

Другой важной частью конфигурации модели является выбор оптимизатора. В свою очередь оптимизатор изменяет установленные вручную параметры. Например, при работе с двумя переменными оптимизатор изучает ландшафт функции, чтобы определить какая скорость обучения лучше всего подходит, чтобы достичь глобальный минимум функции потерь, не перескочив ее слишком большим шагом, но и не устанавливает ее настолько маленькой, что модель никогда не достигнет оптимального значения. Как видно из Таблицы 8 выбор оптимизатора может значительно повлиять на качество модели: при оптимизаторе SGD модели accuracy составляет 0.621, а при RMSprop – 0.813.

Самым популярным оптимизатором на 2020 год является стохастический градиентный спуск (SGD). Градиент считается как сумма градиентов, вызванных каждым элементом обучения. Среднеквадратичное распространение (RMSProp), который поддерживает скорости обучения по каждому параметру. По этой причине оптимизатор хорошо работает на «зашумленных» данных, которыми являются котировки ценных бумаг.Адаптивная оценка момента (Adam) является комбинацией оптимизатора RMSProp с оптимизатором Momentum.

Функция потерь

Функция потерь — функция, которая в теории статистических решений характеризует потери при неправильном принятии решений на основе наблюдаемых данных.

где j – сумма потерь всех предсказанных значений из горизонта, i – сумма потерь по всем компаниям.

Данная функция потерь является одним из самых используемых методов подсчета потерь для задач классификации. Она устойчива к выбросам и подходит для любых задач.

Для данной задачи требуется быстрая модель ввиду работы на высокочастотных данных и ограниченности возможностей для расчетов. Некоторые параметры необходимо настроить путем экспериментов.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Модель 1 | Модель 2 | Модель 3 | Модель 4 | Модель 5 | Модель 6 | Модель 7 |
| num\_layers | 2 | 2 | 2 | 2 | **2** | 1 | 2 |
| dropout | 0.2 | 0.2 | 0.2 | 0.2 | **0.2** | 0.2 | 0.2 |
| batch\_size | 16 | 16 | 16 | 16 | **16** | 16 | 16 |
| learning\_rate | 0.001 | 0.001 | 0.001 | 0.005 | **0.01** | 0.01 | 0.01 |
| learning\_rate\_decay | 0.99 | 0.99 | 0.99 | 0.99 | **0.99** | 0.99 | 0.99 |
| epoch | 3 | 3 | 3 | 3 | **3** | 3 | 3 |
| n\_steps | 10 | 10 | 10 | 10 | **10** | 10 | 5 |
| Optimizer | Adam | SGD | RMSprop | RMSprop | **RMSprop** | RMSprop | RMSprop |
| Accuracy | 0.809 | 0.621 | 0.813 | 0.84 | **0.817** | 0.815 | 0.813 |
| Precision | 0.655 | 0 | 0.664 | 0.704 | **0.716** | 0.658 | 0.646 |
| Recall | 0.805 | 1 | 0.712 | 0.789 | **0.784** | 0.818 | 0.823 |
| Roc-auc | 0.808 | 0.810 | 0.808 | 0.808 | **0.810** | 0.815 | 0.816 |

Таблица 8. Эксперименты по выбору конфигурации модели LSTM. Источник: расчеты автора.

## Метрика качества

Это самая простая метрика качества, но удобная в сравнении моделей. На данный момент это единственная общепринятая метрика, которой можно измерять не бинарные классификации. Она означает количество правильно классифицированных наблюдений по отношению ко всем наблюдениям. Далее будут изучены результаты на валидационной выборке, то есть на тех наблюдениях, на которых не была обучена модель. Так как целью данного исследования не является достижения высокого качества, для определения адекватности модели подойдет случайное гадание в качестве бенчмарка. То есть если модель выдает качество ниже 33,33%, то на основе модели невозможно сделать выводы.

## Классификация

Данные разделяются на 3 класса, которые в последующих экспериментах предсказываются. Разделение происходит на основе цен закрытия этой и следующей минуты. Класс «Повысилась» относится к наблюдениям, у которых цена закрытия следующей минуты выше цены закрытия этой. Классы «Не изменилась» и «Уменьшилась» таким же образом отражают поведение цен закрытия.

где и – это цены закрытия минут t и t+1 соответственно. В целом качественная классификация не обеспечивает прибыли, но это достаточно простая и быстрая модель, с помощью которой можно исследовать и улучшать модель.

## Горизонт предсказания

Были выбраны горизонты предсказания с длинной в 1, 3, 5, 7, 10, 15, 20, 25 и 30 минут. Предсказания длинной в 1 минуту являются самыми простыми и точными, а также самыми популярными в литературе. Далее выбор горизонтов определен компутационными возможностями и достаточно малыми промежутками между ними, чтобы как можно лучше зафиксировать зависимость между исследуемыми показателями.

# Результаты



Таблица 9. Качество моделей с разными временными окнами индикаторов и горизонтами предсказания. Источник: расчеты автора.

Лучшую предсказательную способность имел набор индикаторов с трехминутным временным окном на всех горизонтах предсказания. Таким образом, результаты прошлых работ, где лучшие результаты были, когда длина горизонта предсказания равнялась временному окну технического индикатора. Возможно, это связано с тем, что российский рынок более волатильный и длинные индикаторы имеют низкую предсказательную способность. Также прошлые исследования исследовали дневные цены, а динамика на минутных интервалах может быть совсем иной. Однако результаты данного исследования полностью соответствуют выводу, что именно использование наименьшего временного окна приводит к лучшему качеству модели.

В целом модель на всех горизонтах предсказания кроме 1 минуты показывает низкое качество, но качество предсказания у всех моделей превышает случайное гадание, которое составляет 33,3%. Ввиду чего модель можно признать адекватной, и делать выводы о зависимостях на основе этих результатов.

Любые результаты глубокого обучения очень тяжело интерпретировать, так как модель представляет собой «черную коробку». Ввиду чего невозможно объяснить сравнительно низкие значения в моделях с временным окном индикаторов равным 5 минутам. Это означает, что зависимость не прямолинейная и даже короткие промежутки могут иметь низкое качество. Это определенно требует дополнительного изучения.



Таблица 10. Качество моделей с разными временными окнами индикаторов и горизонтами предсказания в сравнении с моделями без использования технических индикаторов. Источник: расчеты автора.

Тем не менее по Таблице 10 видно, что использование технических индикаторов с маленьким временным окном сильно повышает качество модели на коротких горизонтах предсказания. Например, качество предсказания модели на 1 минуту вперед увеличивается на 59% при добавлении набора технических индикаторов с временным окном 3 минуты. На всех горизонтах модель без технических индикаторов значительно уступает в качестве моделям, включающих их. Однако с увеличением горизонта предсказания добавление индикаторов все меньше улучшает качество модели. Например, при предсказании на 30 значений вперед добавление технических индикаторов улучшает модель лишь на 3,4%.

В целом качество лучшей из построенных моделей составляет 64,39%, что является средним результатом. Но так как целью данного исследования не является улучшение качества, а исследование зависимости, это неважно. На основе полученных результатов и последующих улучшениях возможно построение хорошей модели. На данный момент понятно, что добавление индикаторов значительно улучшает модель, а также что наименьшее временное окно индикатора лучше всего для модели.

# Список литературы

Agrawal M., Khan A. U., Shukla P. K., ‘‘Stock price prediction using technical indicators: A predictive model using optimal deep learning,’’ Learning, vol. 6, no. 2, p. 7, 2019.

Bao, W., Yue, J. and Rao, Y. (2017) ‘A deep learning framework for financial time series using stacked autoencoders and long-short term memory’, PLoS ONE, 12(7), p. 1–24.

Borovkova S., Tsiamas I., ‘‘An ensemble of LSTM neural networks for high-frequency stock market classification,’’ J. Forecasting, vol. 38, no. 6, pp. 600–619, 2019.

Chen K., Zhou Y., Dai F. A lstm-based method for stock returns prediction: A case study of china stock market. In 2015 IEEE International Conference on Big Data (Big Data). IEEE, October 2015.

Chen L., Qiao Z., Wang M., Wang C., Du R., Stanley H.E. Which artificial intelligence algorithm better predicts the chinese stock market? IEEE Access, 6:48625–48633, 2018.

Dezsi E., Nistor I. A. Can deep machine learning outsmart the market? a comparison between econometric modelling and long- short term memory. Romanian Economic Business Review, 11(4.1):54–73, December 2016.

Goodfellow I., Bengio Y., Courville A. (2016). Deep learning. MIT Press.

Hiransha M., Gopalakrishnan E.A., Menon V.K, Soman K.P. Nse stock market prediction using deep-learning models. Procedia Computer Science, 132:1351–1362, 2018.

Hong H., Jeremy C.S., "A unified theory of underreaction, momentum trading, and overreaction in asset markets," The Journal of finance, vol. 54, pp. 2143–2184, December 1999. G. Soros, The alchemy of finance. John Wiley & Sons, 2015.

Kara Y., Boyacioglu M. Acar, Baykan O.K. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange Expert Systems with Applications, 38 (5) (2011), pp. 5311-5319

Keynes J.M., The general theory of employment, interest, and money. Macmillan, London, 1936. Neely C.J., Rapach D.E., Jun T., Guofu Z., "Forecasting the Equity Risk Premium: The Role of Technical Indicators." Management Science. vol. 60, pp. 1772–1791, July 2014.

Khare K., Darekar O., Gupta P., Attar V. Z. Short term stock price prediction using deep learning. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT). IEEE, May 2017.

Moews, B. and Ibikunle, G. (2020) ‘Predictive intraday correlations in stable and volatile market environments: Evidence from deep learning’, Physica A: Statistical Mechanics and its Applications, p. 547.

Mnih V., Badia A. P., Mirza M., Graves A., Lillicrap T. P., T. Harley, D. Silver, and K. Kavukcuoglu, “Asynchronous methods for deep reinforcement learning,” in Proc. Int. Conf. on Machine Learning, New York City, USA, June 19 –24, 2016, p. 1928−1937.

Nelson D. M. Q., Pereira A. C. M., de Oliveira R. A., ‘‘Stock market’s price movement prediction with LSTM neural networks,’’ in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Anchorage, AK, USA, May 2017, pp. 1419–1426.

Ponomarev, E. S., Oseledets, I. V. and Cichocki, A. S. (2019) ‘Using Reinforcement Learning in the Algorithmic Trading Problem’, Journal of Communications Technology and Electronics, 64(12), p. 1450.

Selvin S., Vinayakumar R., Gopalakrishnan E. A., Menon V.K, Soman K. P. Stock price prediction using lstm, rnn and cnn-sliding window model. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, September 2017.

Sezer O.B., Gudelek U., Ozbayoglu M. Financial time series forecasting with deep learning: A systematic literature review: 2005-2019, Applied Soft Computing, 90 (2020), Article 106181.

Shynkevich Y., McGinnity T.M., Coleman S.A., Belatreche A., Li Y. Forecasting price movements using technical indicators: investigating the impact of varying input window length. Neurocomputing. 2017; p. 71-88.

Tan X., Li S., Wang C., Wang S., ‘‘Enhancing high frequency technical indicators forecasting using shrinking deep neural networks,’’ in Proc. 6th Int. Conf. Inf. Manage. (ICIM), Mar. 2020, pp. 249–256.

Treynor J. L., Ferguson R. (1985) ‘In Defense of Technical Analysis’, Journal of Finance (Wiley-Blackwell), 40(3), pp. 757–773. doi: 10.1111/j.1540-6261.1985.tb05000.x.

Zhou X., Pan Z., Hu G., Tang S., Zhao C., ‘‘Stock market prediction on high-frequency data using generative adversarial nets,’’ Math. Problems Eng., vol. 2018, pp. 1–11, Apr. 2018.