МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Кафедра дискретной математики и алгоритмики

ШУР Наталья Александровна

ПРОГНОЗИРОВАНИЕ ЦЕН ФЬЮЧЕРСНЫХ КОНТРАКТОВ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Магистерская диссертация
специальность 1-31 81 09 «Алгоритмы и системы
обработки больших объемов информации»

Научный руководитель
Юрий Леонидович Орлович
кандидат физ.-мат. наук

Допущена к защите
«___»_____ 2020 г.
Зав. кафедрой дискретной математики и алгоритмики
_____ В. М. Котов
доктор физ.-мат. наук, профессор

Минск, 2020

ОГЛАВЛЕНИЕ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Магистерская диссертация, 42 с., 22 источника.

ПРОГНОЗИРОВАНИЕ ЦЕН, КНИГА ЗАЯВОК, НЕЙРОННАЯ СЕТЬ, МАШИННОЕ ОБУЧЕНИЕ, ОТБОР ПРИЗНАКОВ

Объект исследования — задача прогнозирования цен фьючерсных контрактов по данным книги заявок, алгоритмы на основе нейронных сетей для решения исследуемой задачи, а также извлечение признаков из книги заявок.

Цель работы — исследование методов на основе нейронных сетей для прогнозирования цен фьючерсных контрактов; изучение существующих подходов, их усовершенствование и поиск новых решений.

В ходе работы были исследованы существующие нейросетевые модели для решения поставленной задачи и подходы к извлечению признаков для обучения моделей. Был проведен сравнительный анализ методов и рассмотрены способы их усовершенствования. Также были рассмотрены ранее не применявшееся для обучения на данных книги заявок подходы.

Актуальность магистерской диссертации обусловлена отсутствием общепринятого и эффективного подхода к решению поставленной задачи. Описанные в литературе методы нельзя использовать в исходном виде для решения практических задач, например построения алгоритмов автоматического трэйдинга. Результативность методов также очень зависит от источника данных, интенсивности торговли на бирже, времени. Поэтому для каждой индивидуальной практической задачи необходимо проводить отдельное исследование, которое становится значительно проще, если существует набор подходов с эвристическими предположениями, подтвержденными экспериментом, о свойствах данных, на которых эти подходы показывают хорошее качество. Именно такой набор описывается и исследуется в данной работе.

Диссертация состоит из разделов «Введение», «Общая характеристика работы», «Заключение» и «Список использованных источников» и основной части. Основная часть включает 4 главы. В первых двух главах, «Искусственные нейронные сети для прогнозирования финансовых временных рядов» и «Способы задания книги заявок для обучения нейронных сетей», описываются методы и подходы в общем виде. Последующие главы «Описание данных» и «Обучение и тестирование» включают описание данных и экспериментов, а также подробное описание результатов.

АГУЛЬНАЯ ХАРАКТЕРЫСТЫКА РАБОТЫ

Магістарская дысертацыя, 42 с., 22 крыніцы.

ПРАГНАЗАВАННЕ ЦЭН, КНІГА ЗАЯВАК, НЕЙРОННАЯ СЕТКА, МАШЫННАЯ НАВУЧАННЕ, АДБОР ПРЫЗНАКАЎ

Об'ект даследвання — задача прагназавання цэн фьючарсных кантрактаў по дадзеным кнігі заявак, алгарытмы на основе нейронных сетак для рашэння даследуемай задачы, а таксама адбор прызнакаў з кнігі заявак.

Мэта работы — даследванне метадаў на базе нейронных сетак для прогназавання цэн фьючарсных кантрактаў; вывучэнне існуючых подыходаў, іх ўдасканалення і пошук новых рашэнняў.

Падчас працы даследваліся існуючыя нейрасецявые маделі для рашэння пастаўленная задачы і подыходы да адбору прызнакаў для навучання алгарытмаў. Был праведзены параўнальны аналіз метадаў і разгледжаны спосабы их удасканалання. Таксама раглядадліся раней не ужываныя для навучання на дадзеных кнігі заявак подыходы.

Актуальнасць магістэрскай дысертацыі абумоўлена адсутнасцю агульнапрынятага і эфектыўнага падыходу да вырашэння пастаўленай задачы. Апісаныя ў літаратуры метады нельга выкарыстоўваць у зыходным выглядзе для рашэння практычных задач, напрыклад пабудовы алгарытмаў аўтаматычнага трэйдзінгу. Выніковасць метадаў таксама вельмі залежыць ад крыніцы дадзеных, інтэнсіўнасці транзакцый на біржы, часу. Таму для кожнай індывідуальнай практычнай задачы неабходна праводзіць асобнае даследаванне, якое становіцца значна прасцей, калі існуе набор падыходаў з эўрыстычнымі здагадкамі, пацверджанымі эксперыментам, пра ўласцівасці дадзеных, на якіх гэтыя падыходы паказваюць добрую якасць. Менавіта такі набор апісваецца і даследуецца ў дадзенай працы.

Дысертацыя складаецца з раздзелаў «Уводзіны», «Агульная характарыстыка работы», «Заключэнне» і «Спіс выкарыстаных крыніц» і асноўнай часткі. Асноўная частка складаецца з 4 частак. У першых дзвюх частках, «Штучныя нейронавыя сеткі для прагназавання фінансавых часовых шэрагаў» і «Спосабы задання кнігі заявак для навучання нейронных сетак», апісваюцца метады і падыходы ў агульным выглядзе. Наступныя разделы «Апісанне дадзеных» і «Навучанне і тэставанне» ўключаюць апісанне дадзеных і эксперыментаў, а таксама падрабязнае апісанне вынікаў.

ABSTRACT

Master's thesis, 42 pp., 22 Sources.

PRICE FORECASTING, ORDER BOOK, NEURAL NETWORK, MACHINE LEARNING, FEATURE SELECTION

Object of the research — price forecasting problem for future contracts by the order book data, algorithms based on neural networks to solve the problem, and the extraction of features from the order book.

The purpose of the work — the study of methods based on neural networks to predict the prices of futures contracts; the study of existing approaches, their improvement and new solutions development.

In the course of the work, the existing neural network models for the problem and approaches to the features selection for models training were studied. A comparative analysis of the methods was carried out and ways to improve them were considered. Some approaches that were not used for order book data snapshots yet were also considered.

The relevance of the master's thesis can be explained by the lack of a well–known and effective approach to solving the problem. The methods described in the literature cannot be used in their original form in applications, e. g. constructing algorithms for automatic trading. The effectiveness of the methods is also very dependent on the data source, the intensity of trading on the exchange, time. Therefore, for each individual practical case, it is necessary to conduct a separate study. And it becomes much simpler if there is a set of approaches with heuristic assumptions confirmed by experiment on the properties of the data on which these approaches show good quality. Like the set described and investigated in this paper.

The dissertation consists of sections «Introduction», «General characteristics of the work», «Conclusion» and «List of sources used» and the main part. The main part includes 4 chapters. The first two chapters, «Artificial Neural Networks for Predicting Financial Time Series» and «Ways to Define Order Book for Learning Neural Networks», describe methods and approaches in a general way. The subsequent chapters «Data Description» and «Training and Testing» include a description of the data and experiments, and a detailed description of the results.

ВВЕДЕНИЕ

Сегодня все основные фондовые биржи предоставляют сервисы для автоматической торговли, и в частности, для высокочастотного трейдинга (HFT — High Frequency Trading). Как правило торговля осуществляется по некоторому алгоритму, реализуемом в виде компьютерной программы. Принято алгоритм (иногда систему в целом) называть стратегией.

В процессе высокочастотной торговли генерируются колоссальные объемы данных. Состояние книги заявок (Order Book) может меняться с частотой в десятки обновлений в наносекунду, как и количество сделок (трейдов). Соответственно, финансовые данные, как правило, представляют собой временные ряды.

Зачастую основой торговой стратегии является прогнозирование стоимости финансовых активов и других производных показателей в том или ином виде. От эффективности решения данной задачи зависит получаемая финансовая выгода и подверженность алгоритма сопутствующим рискам.

Сложность работы с высокочастотными данными при решении задачи прогнозирования заключается в интерпретации влияния одного события (новая заявка (ордер), трейд, отмена ордера и т. д.) на целевую (предсказываемую) величину. Как правило, для исследования используются последовательные снимки (snapshots) книги заявок — вектора, содержащие количества ордеров на покупку и продажу по ценовым уровням. Для решения задач краткосрочного прогнозирования с данными такого типа зачастую используются методы машинного обучения. В работе предполагается рассмотрение данных методов, их сравнение и способы улучшения их точности предсказания.

Определения и базовые понятия

Для обозначения цены ордеров на покупку будет использовано обозначение *bid*, а на продажу — *ask* (или *offer*), с указанием уровня цены или ее абсолютной величины. Лучшие цены продажи и покупки на рынке в некоторый момент могут называться просто bid и ask, соответственно. Эту пару значений принято обозначать BBO (от Best Bid Offer). *Mid–цена* — среднее значение bid и ask. *Спред* (spread) — разность между ask и bid (либо разность между некоторыми показателями на определенных уровнях книги; значение

в каждом конкретном случае определяется из контекста).

Заявку на продажу или покупку будем называть также *ордер (order)*.

Лимитный (или отложенный) ордер (limit order) — это тип ордера на покупку (или продажу) актива по цене не выше (или не ниже) указанной цены, соответственно. *Рыночный ордер (market order)* — это тип ордера, который исполняется сразу по лучшей доступной на рынке цене.

Ордер является *агрессивным*, если он исполняется сразу: его цена на продажу (покупку) не больше (меньше) лучшей цены на покупку (продажу) на рынке. Напротив, *пассивные* ордера попадают в книгу заявок и остаются там до изменения состояния рынка или до их отмены трейдером или биржей.

Заметим, что пассивным ордер может быть (в наших определениях и в предположении, что мы не рассматриваем другие типы ордеров) только в случае, если он был послан как лимитный.

В работе будет рассмотрена *книга отложенных заявок (limit order book, LOB)*, которая подразумевает пассивные ордера, которые были созданы как отложенные (limit orders).

Все термины, относящиеся к фондовым рынкам, используемые, но не определенные в данной и последующих главах, можно найти в [4].

ГЛАВА 1
КНИГА ЗАЯВОК. ПОСТАНОВКА ЗАДАЧИ ПРОГНОЗИРОВАНИЯ

1.1 Книга заявок

	Price	**Ask Size (MBO)**
	10.42	10
	10.41	1 10
	10.40	1 1 10
	10.39	5 12
	10.38	1 1 2 5 2
2 4 4 1	10.36	
1 2 1	10.35	
10 2 1 1 1	10.34	
3 11	10.33	
23	10.32	
Bid Size (MBO)	**Price**	

Таблица 1.1: Пример. Limit order book

Книгу заявок можно описать как таблицу, содержащую все заявки на рынке, сгруппированные по ценовым уровням.

Для цены на некоторый актив устанавливается *минимальный инкремент (тик, tick size)*, и корректная цена для ордера должна быть кратна этому значению (например, 0.01 в примере представленном таблицей 1.1).

Уровень (price level) — некоторая цена и все заявки по этой цене на рынке. Как правило берут фиксированное количество ценовых уровней (в примере 5), начиная от BBO в сторону убывания цены для bid и в сторону возрастания — для ask.

Размер ордера (order size, quantity) — количество контрактов (или других единиц), которое хотят купить или продать. Книги классифицируются по

тому, как в них представлена информация о заявках на уровне. Если каждый ордер представлен в отдельности — это *Market by Order (MBO)* книга, если же дан только суммарный размер ордеров для каждого уровня — это *Market by Price (MBP)* книга (пример в таблице 1.2).

	Price	**Ask Size (MBP)**
	10.42	10
	10.41	11
	10.40	12
	10.39	17
	10.38	11
11	10.36	
4	10.35	
15	10.34	
14	10.33	
23	10.32	
Bid Size (MBP)	**Price**	

Таблица 1.2: Пример. MBP книга для MBO в таблице 1.1

Уровни в книге могут не содержать заявок (быть пустыми). Также, как показано в примерах, разница BBO (спред) может быть больше одного тика. Для прогнозирования часто берут mid-price: среднее bid и ask.

На торговых площадках функционируют различные алгоритмы исполнения ордеров. Наиболее распространенными являеются: first–in–first–out и pro–rata.

1.2 Постановка задачи

Задача прогнозирования может быть математически формализована следующим образом.

Пусть X_t — набор базовых индикаторов (например, "снимки" книги заявок) и Y_t — предсказываема величина в момент времени $t = 1, 2, ..., T$. Необходимо, имея исторические данные $X_t = \{X_1, X_2, ..., X_T\}$ и соответствующие

$Y_t = \{Y_1, Y_2, ..., Y_T\}$, сделать прогноз значения Y_{T+1} в следующий момент наблюдения.

При построении торговых алгоритмов предсказываемой величиной может быть как непрерывная величина (часто условно, так как цена финансовых активов, например, все же дискретна), так и величина, принимающая конечное множество значений. Например, направление изменения величины показателя. То есть пусть на ряду с базовыми показателями у нас имеется $d_t = \phi(Y_t - Y_{t-1}), t = 2, 3, ..., T, \phi : \mathbb{R} \to D, |D| < \infty$. Прогнозируемое значение в этом случае — d_{T+1}.

Таким образом имеем две постановки проблемы: первая — задача регрессии, вторая — задача классификации. В данной работе основное внимание будет уделено второй задаче.

Описанные задачи могут быть рассмотрены как традиционные проблемы предсказания временных рядов. Но все же большинство исследователей выделяют задачу прогнозирования для высокочастотного трейдинга, как особенную из–за специфики рыночных данных.

ГЛАВА 2
ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ ДЛЯ ПРОГНОЗИРОВАНИЯ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ

2.1 Обзор архитектур искусственных нейронных сетей для прогнозирования финансовых временных рядов

Зачастую алгоритмы в высокочастотном трейдинге подразумевает получение прибыли на малых и частых изменениях на рынке. Объемы данных, возникающие при этом, и их многомерная природа приводят исследователей к использованию искусственных нейронных сетей.

Рекуррентные сети, как хорошо зарекомедовавшие себя в прогнозировании временных рядов в целом, широко применяются и для прогнозирования [13] и классификации высокочастотных финансовых временных рядов [3].

Одним из самых популярных решений является рекуррентная нейронная сеть «Долгая краткосрочная память» (Long short-term memory (LSTM)) [2, 22] и её разновидности.

Для извлечения «пространственных» признаков из книги заявок применяются сверточные нейронные сети или отдельные слои [18, 22, 14].

Также, в некоторых работах используются нейронные сети прямого распространения, например [7] для решения задачи прогнозирования и задачи детектирования алгоритмического трейдинга в [1].

Все чаще появляются работы по прогнозированию цен акций, в которых используются генеративно–состязательные сети (GANs) [19, 21] и обучение с подкреплением (RF) [20].

На основании данных статей по теме были выбраны несколько архитектур для тестирования. Их подробное описание будет предложено в следующей главе. Далее будут рассмотрены типы архитектур, являющиеся базовыми для выбранных моделей.

2.1.1 Сверточные нейронные сети

Сверточные нейронные сети (Convolutional neural network, CNN) широко применяются для анализа изображений.

Сверточная нейронная сеть может состоять из одного или нескольких блоков, которые в свою очередь включают: сверточный слой, слой активации и слой пулинга 2.1.

Сверточный слой — базовый слой сети.

Формально свертку f и g можно определить так

$$(f * g) = \sum_{k,l} f(m-k, n-l) \cdot g(k,l).$$

Для вычисления свертки *ядро свертки* g сдвигают относительно предыдущего слоя f, почленно перемножая и складывая.

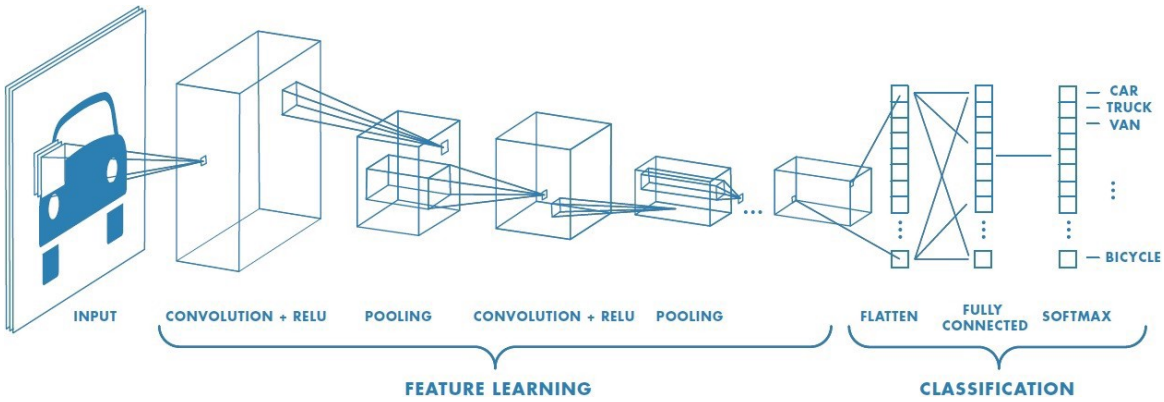


Рисунок 2.1: Типичная структура сверточной нейронной сети (источник https://towardsdatascience.com/)

Слой активации — некая нелинейная функция. Классическими уже можно назвать гиперболический тангенс ($f(x) = \tanh(x)$) и сигмоиду ($f(x) = (1 + e^{-x})^{-1}$. Наиболее популярна из–за сравнительной простоты вычисления и эффективности в решении проблемы «затухания градиента» функция активации ReLu ($f(x) = max(0, x)$) и её разновидности.

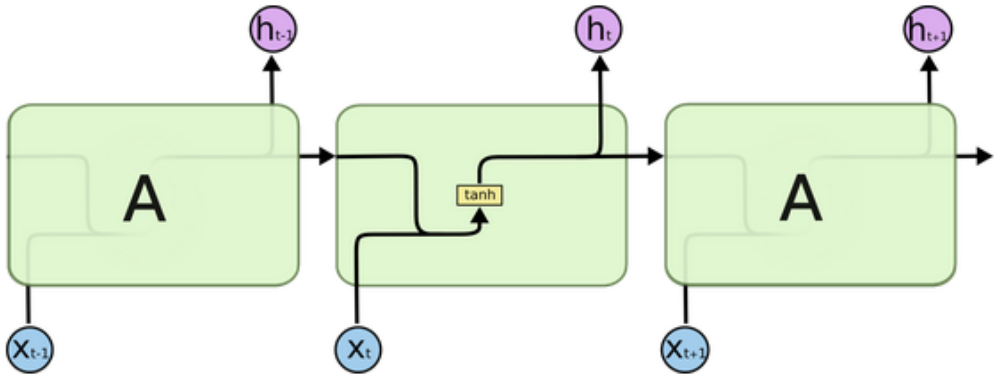
Слой пулинга (подвыборки) — нелинейное уплотнение карты признаков, что подразумевает «уплотнений» группы пикселей до одного пикселя, через некоторое нелинейное преобразование (например, функция максимума или среднее арифметическое).

Операция свертки может быть интерпретирована, как извлечение более общих, «высокоуровневых» признаков с каждым слоем.

2.1.2 Долгая краткосрочная память (LSTM)

Долгая краткосрочная память (Long short-term memory, LSTM) — одна из разновидностей архитектуры рекуррентных нейронных сетей, основной особенностью которой является способность к обучению долговременным зависимостям [5].

Рекуррентные нейронные сети имеет форму цепочки повторяющихся модулей. В самом простом случае один такой модуль может представлять собой единственный слой с функцией активации tanh (рис. 2.2).



The repeating module in a standard RNN contains a single layer.

Рисунок 2.2: Структура рекуррентной нейронной сети (источник https://towardsdatascience.com/)

Соответствующий модуль LSTM имеет более сложную структуру и, как правило, состоит из «четырех слоев» (рис. 2.3).

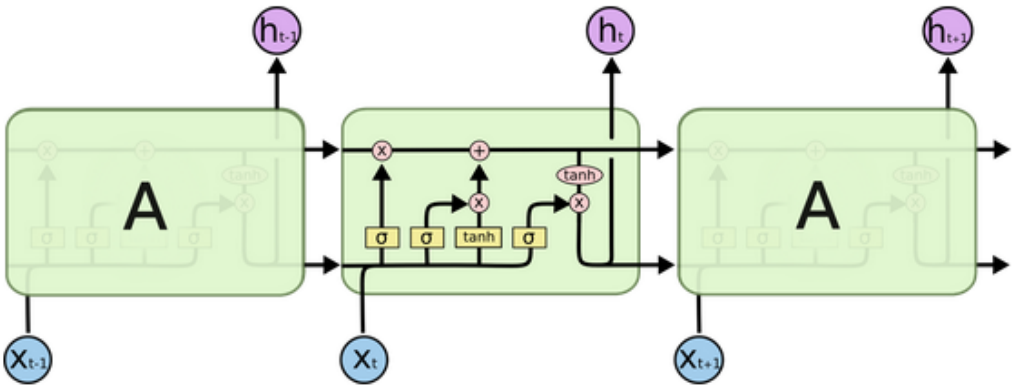


Рисунок 2.3: Структура «долгой краткосрочной памяти» (LSTM) (источник https://towardsdatascience.com/)

Классический LSTM модуль можно описать с помощью следующей системы:

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t),
\end{aligned}
$$

где $\circ$ — обозначает произведение Адамара,

x_t — входной вектор, h_t — выходной вектор ($h_0 = 0$),

c_t — вектор состояний ($c_0 = 0$),

W, U и b — матрицы параметров и вектор,

f_t, i_t и o_t — векторы «вентилей» (веса запоминания старой информации, получения новой информации, и "степень уверенности" в кандидате на выходное значение),

а также функции активации:

σ_g — на основе сигмоиды,

σ_c — на основе гиперболического тангенса,

σ_h — на основе гиперболического тангенса (в другой разновидности LSTM (LSTM со «смотровыми отверстиями») предполагается, что $\sigma_h(x) = x$.

2.2 Генеративно–состязательные сети

Генеративно–состязательная сеть (Generative Adversarial Net, GAN) впервые была представлена Иэном Гудфеллоу в 2014 году. В классическом варианте это алгоритм машинного обучения, построенный на комбинации из двух нейронных сетей: генеративной модели G, которая строит приближение распределения данных, и дискриминативная модель D, оценивающая вероятность, что образец пришел из тренировочных данных, а не из сгенерированных моделью G. Обучение для модели G заключается в максимизации вероятности ошибки дискрминатора D.

Чаще всего GAN используют для генерации изображений.

Однако эту архитектуру в несколько измененном виде также успешно применяли и в предсказании временных рядов [19, 21]. В качестве генерируемого объекта выступает продолженный временной ряд. Как правило, в качестве генератора берется сеть, основанная на LSTM, а в качестве дискриминатора — некоторая сверточная сеть.

2.3 Описание исследуемых в работе моделей

Для отрисовки диаграмм использовался визуализатор Netron (`https://github.com/lutzroeder/netron`).

2.3.1 Simple LSTM

В качестве отправной точки и для тестирования установленного процесса обучения и тестирования использовалась сеть на основе LSTM (рис. 2.4).

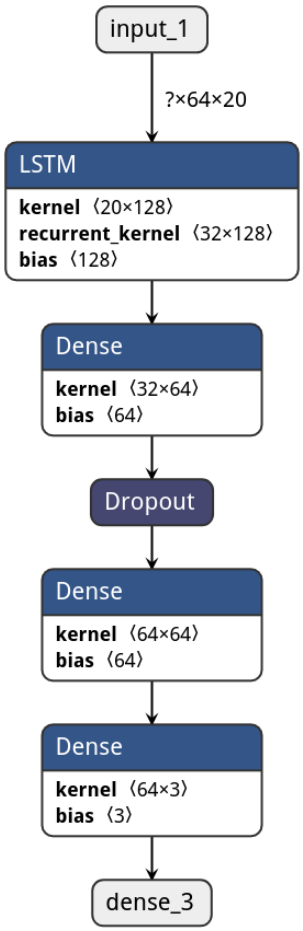


Рисунок 2.4: Simple LSTM

Далее в работе она будет упоминаться как "Simple LSTM" .

2.3.2 CNN

Как упоминалось ранее сверточные нейронные сети используются для извлечения признаков из книги заявок [17, 22, 18] (рис. 2.5).

Данная модель повторяет аналогичную в [18].

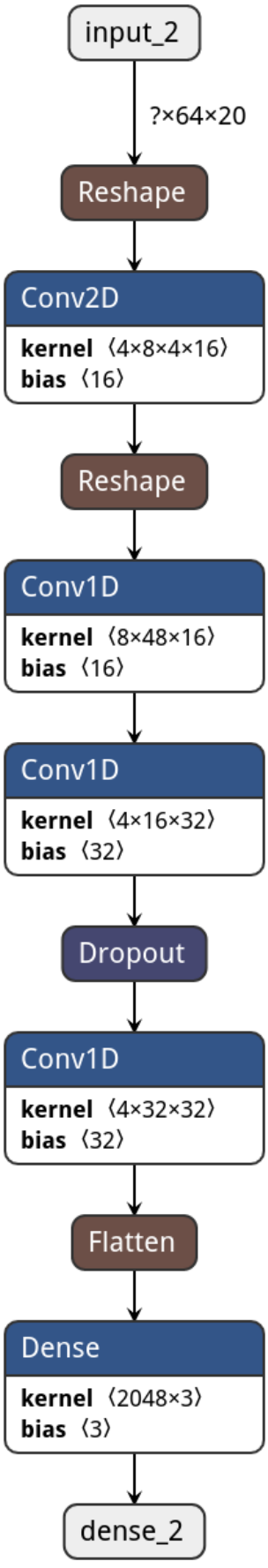


Рисунок 2.5: CNN

2.3.3 ConvLSTM

Convolutional LSTM — гибридный вариант. На вход рекуррентной сети подается информация уже более «высокого» уровня со сверточных слоев (рис. 2.6).

Данная модель повторяет аналогичную в [18].

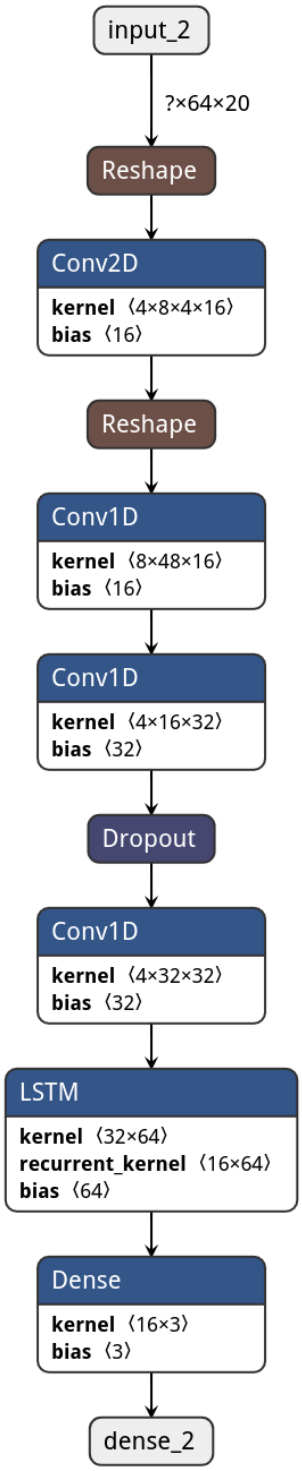


Рисунок 2.6: ConvLSTM

2.3.4 DeepLOB

Данную архитектуру предлагают авторы статьи [22].

Также гибридный вариант, комбинирующий сверточные и рекуррентные сети (рис. 2.7). Отличительной чертой этой модели является использование так называемого *Inception Module* [15]).

Будучи впервые успешно использован для глубокого обучение в соревновании ImageNet, данный модуль оказывается полезным и во многих других задачах. Авторы статьи [22] указывают, что его использование позволяет сети «ловить» паттерны по разным временным (событийным) шкалам, не ограничиваясь кратными соответствующим размерам сверток. Также использование данного модуля можно интерпретировать как применение скользящего среднего с различными размерами окон в техническом анализе.

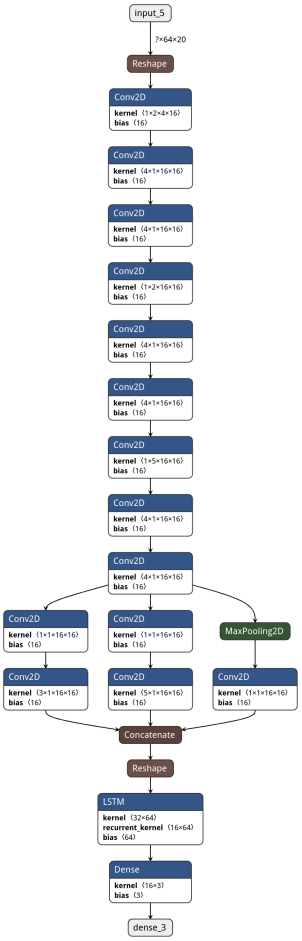


Рисунок 2.7: DeepLOB

2.3.5 GAN–FD

GAN–FD (GAN for minimizing forecast error loss and direction prediction loss) — GAN для минимизации ошибки предсказания и ошибки направления движения цены [19]. В качестве генератора в этой сети используется LSTM, а в качестве дикриминатора — сверточная сеть.

В статье данная модель применялась для предсказания цен закрытия акций. Как показывают результаты, данная архитектура способна улучшать результаты модели–генератора.

В данной работе генератором была выбрана ConvLSTM, дискриминатором — сверточная сеть с тремя скрытыми слоями аналогичная архитектуре, названной в данной работе CNN.

На вход генератору подавались обработанные исходые данные, входом дискриминтора были также исходные данные и предсказание генератора. Сверточная часть дискриминатора использовалась для преобразования временного ряла из исходных данных, далее выход сверточной части конкатенировался с предсказанием и снова с помощью двух сверточных слоев и одного полносвязного слоя получали предсказание дискриминатора.

В качестве функции потерь использовалась аналогичная [19].

ГЛАВА 3
СПОСОБЫ ЗАДАНИЯ КНИГИ ЗАЯВОК ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ

Для обучения нейронных сетей используют как снимки книги заявок непосредственно, так и некоторые производные варианты. В данной главе кратко описаны некоторые изветные способы задания книги заявок, а также предложен новый способ («дельта–признаки»).

3.1 Необработанные данные

Самым простым сосом представления книги заявок является снимок книги заявок «как есть» .

Структура одного снимка в общем виде представляется как:

$$\{p_a^{(i)}, v_a^{(i)}, p_b^{(i)}, v_b^{(i)}\}_{i=1}^{n},$$

где $p_a^{(i)}, v_a^{(i)}$ — цена и объем i–ого уроня в книге на продажу и $p_b^{(i)}, v_b^{(i)}$ — цена и объем i–ого уровня в книге на покупку соответственно, а n — количество уровней в книге (часто 2, 5 или 10).

Также в снимок могут быть включены такие характеристики как $n_b^{(i)}$ и $n_a^{(i)}$ — количество заявок на соответствующем уровне.

Данный способ является также самым быстрым и предпочтительным для реализации: при принятии решений онлайн не требуется никакой дополнительной обработки данных.

3.2 Стационарные признаки

На вход сети подается не снимок книги непосредственно, а некоторые признаки (Stationary features [18]), вычисленные по исходным данным.

Определим

$$p_m(t) = \frac{p_a^{(1)}(t) + p_b^{(1)}(t)}{2}.$$

В таблице 3.1 кратко описаны признаки указанные в [18]. В данной работе

использование этих признаков улучшило результаты моделей, основанных на нейронных сетях.

Преобразование цен на уровнях обосновываются необходимостью последовательной нормализации данных. Если в случае с объемом на уровне проблем нет, то с ценой они возникают, так как цена меняется динамически. И по всему набору данных нормализацию провести корректно не всегда представляется возможным.

В качестве «компенсации» удаления обсолютных значений цен преобразованием, упомянутым выше, предлагается использовать изменение средней цены по сравнению с предыдущим снимком книги в качестве признака.

Кумулятивный объем можно объяснить как объем, который необходимо «сторговать» , чтобы текущий уровень стал лучшим. Также по ней можно судить о «тяжести» данной стороны книги.

Таблица 3.1: Стационарные признаки (Stationary features) [18]

Признак	**Определение**
Разница ценовых уровней	$p'^{(i)}(t) = \frac{p^{(i)}(t)}{p_m(t)} - 1$
Изменение цены	$p_m'^{(i)}(t) = \frac{p^{(i)}(t)}{p_m(t-1)} - 1$
Кумулятивный объем	$v'^{(k)}(t) = \sum_{i=1}^{k} v^{(i)}(t)$

В работе исследовался этот подход, однако с небольшими изменениями: изменения цен считались не относительно средней цены, а в тиках (минимально возможном изменении цены).

3.3 Дельта–признаки

В данном подходе предлагается использовать кроме кумулятивного абсолютного значения объема на уровне, «дельту» — разницу между текущим и предыдущим на соответствующих уровнях:

$$\Delta v^{(i)}(t) = v^{(i)}(t) - v^{(i)}(t-1).$$

При изменении лучшей цены возможно, что предыдущего уровня нет — появился новый уровень с ценой лучше предыдущей, либо наоборот — удален

уровень.

В первом случае предыдущий снимок сдвигается в сторону увеличения номеров уровней и в качестве объема на первом уровне добавляется «0» . Во втором — сдвигается текущий снимок также в сторону увеличения номеров уровней и снова добавляется «0» вначало.

Эффект, который достигается данным преобразованием не может быть достигнут простой сверткой из–за того, что в данные не выравнены по ценовым уровням.

На рисунках 3.1 – 3.3 показаны данные преобразования для разных случаев.

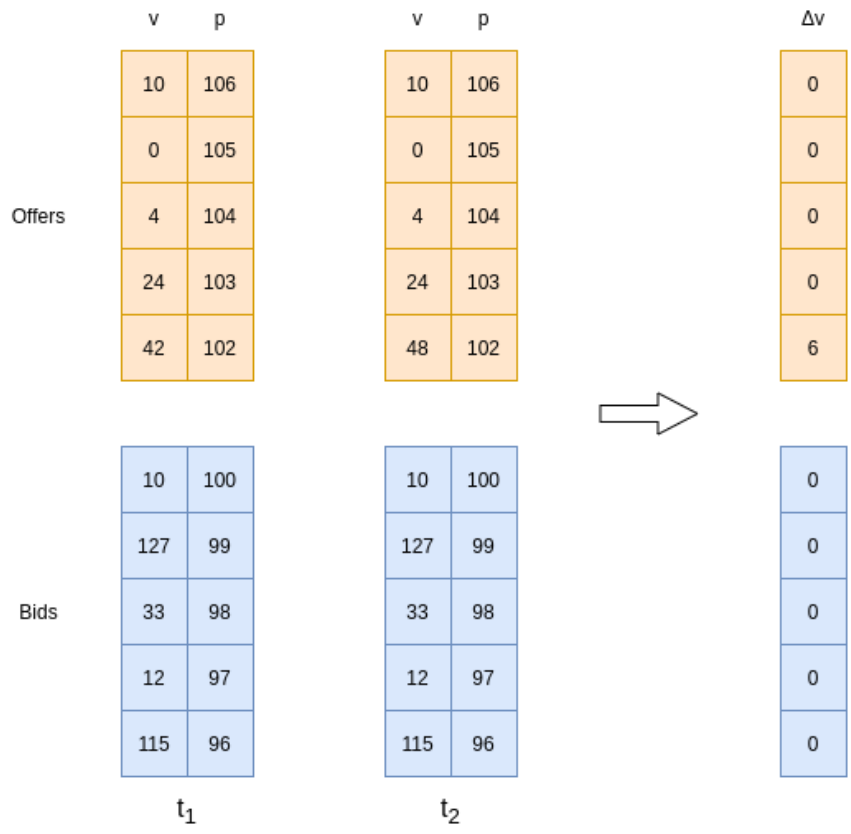


Рисунок 3.1: Пример вычисления Δv. Цена не изменяется

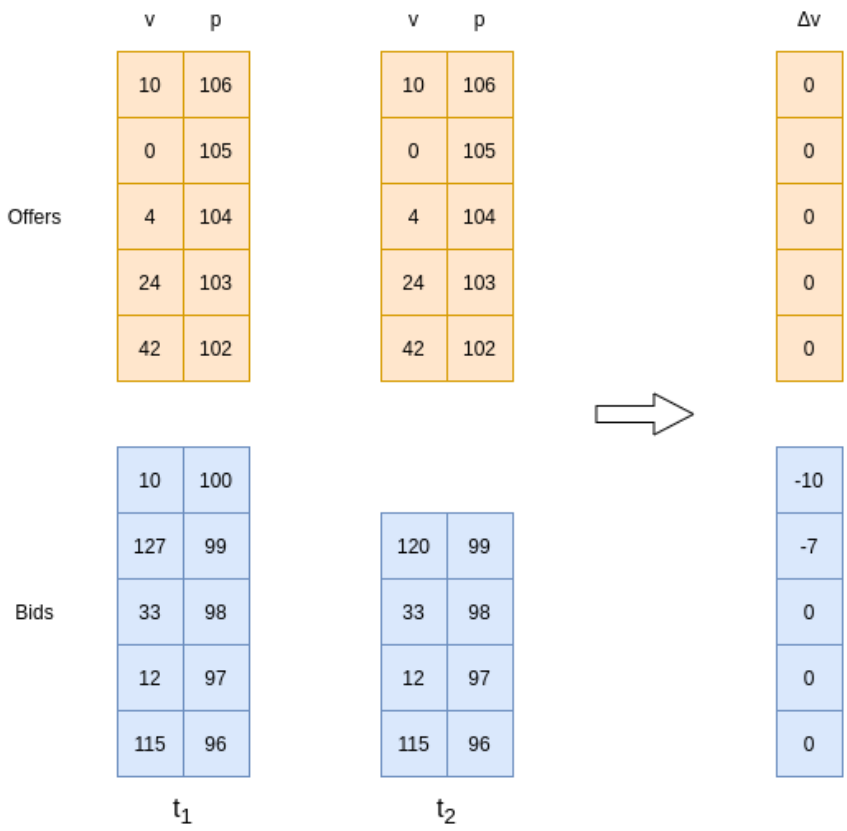


Рисунок 3.2: Пример вычисления Δv. Удален уровень

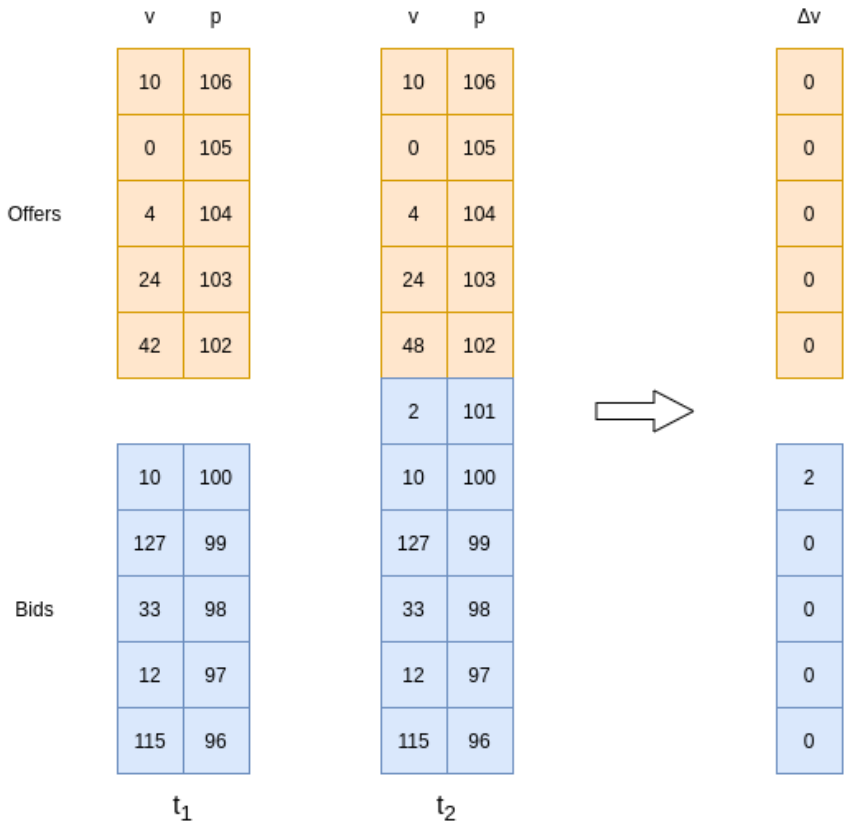


Рисунок 3.3: Пример вычисления Δv. Добавлен уровень

В пробных экспериментах добавление признака «Измененинние цены» из Stationary features не улучшало качество на тестовой выборке, поэтому в этот набор признаков он не включен.

Преобразования цен на уровнях аналогично соответствующему в Stationary features. Но для того, чтобы все же добавить в данные величину, характеризующую движение цены на данном временном ряде, относительное значение выщитывалось от mid–price с предыдущего снимка.

$$p'^{(i)}(t) = \frac{p^{(i)}(t)}{p_m(t-1)} - 1$$

Как уже упоминалось ранее признак «Кумулятивный объем» из Stationary features был использован в исходном виде, так как данный признак является важной характеристикой книги, влияющей в том числе на движение цены.

Кратко все дельта–признаки приведены в таблице 3.2.

Таблица 3.2: Дельта–признаки (Delta features)

Признак	**Определение**
Разница ценовых уровней	$p'^{(i)}(t) = \frac{p^{(i)}(t)}{p_m(t-1)} - 1$
Разница объемов	$\Delta v'^{(i)}(t) = v^{(i)}(t) - v^{(i)}(t-1)$
Кумулятивный объем	$v'^{(k)}(t) = \sum_{i=1}^{k} v^{(i)}(t)$

К данному подходу справедливо также замечание о признаке «Разница ценовых уровней»: изменения цен считались не относительно средней цены, а в тиках (минимально возможном изменении цены).

ГЛАВА 4
ОПИСАНИЕ ДАННЫХ

4.1 Структура набора данных

Набор данных представляет собой «снимки» книги заявок после каждого публикуемого события на бирже. Он содержит данные с 6 по 31 января 2019 года по фьючерсному контракту S&P 500 с датой исполнения в марте 2019 года и включает более 1.4 миллиона записей.

Структура одной записи:

$$\{p_a^{(i)}, v_a^{(i)}, p_b^{(i)}, v_b^{(i)}\}_{i=1}^{n}, n = 5$$

где $p_a^{(i)}, v_a^{(i)}$ — цена и объем i–ого уроня в книге на продажу и $p_b^{(i)}, v_b^{(i)}$ — цена и объем i–ого уроня в книге на покупку соответственно.

В качестве величины для предсказания зачастую используется [10, 6] индикатор отклонения цены через k событий:

$$r_i = \frac{p_m^{(i+k)} - p_m^{(k)}}{p_m^{(k)}}$$

$$y_i = \begin{cases} -1, r_i < -\alpha, \\ 0, -\alpha \leq r_i \leq \alpha, \\ 1, r_i > \alpha, \end{cases}$$

где $p_m(t)$ — mid–price, определяемый по следующей формуле

$$p_m(t) = \frac{p_a^{(1)}(t) + p_b^{(1)}(t)}{2},$$

а α — фиксированный порог «значимости» изменения.

Из–за сильного шума в данных для вычисления индикатора в качестве p_m берут среднее значени mid–price за определенное количество событий.

В данной работе использовался измененный индикатор:

$$r_i^* = \frac{p_m^{(i+k)} - p_m^{(k)}}{t},$$

t — размер минимального инкремента для инструмента. В экспериментах пороговое значение $\alpha = 0.9$ (фактически соответствует движению в 1 тик), усреднение цены проводилось по 10 событиям.

Число k называю «горизонтом» прогноза. В экспериментах данной работы $k = 20$, что соответствует 20–100 мкс.

В качестве признаков использовались 64 предыдущих снимков книги заявок.

4.2 Протокол экспериментов

Исходный датасет был разделен на три части: обучающую, валидационную и тестовую выборки. Для обучения использовались первые две недели (около 700 тыс. снимков), следующая неделя — для валидации и последняя — для тестирования (более 300 тыс. снимков в каждой).

Средняя точность в силу несбалансированности выборки (почти 85% — класс «0» , рис. 4.1) не может отражать в полной мере результаты экспериментов.

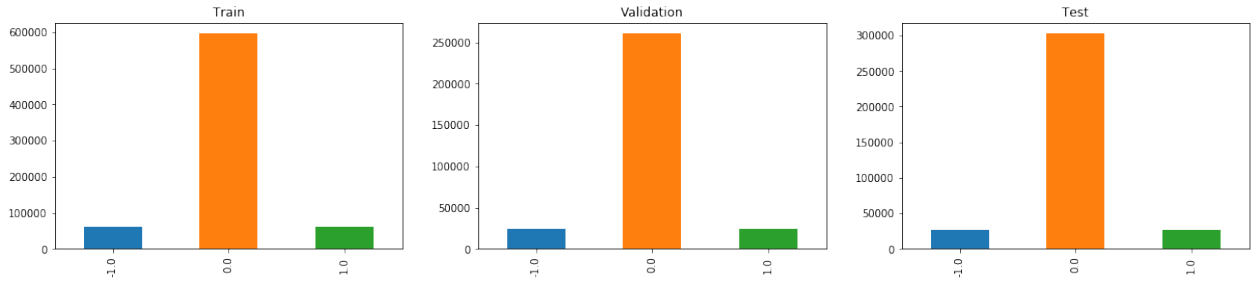


Рисунок 4.1: Распределение прогнозируемой величины по классам

Поэтому в качестве метрик предлагается использовать кроме средней точности (mean accuracy), полноту (recall), точность (precision) и $F1$–меру (F1–score).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{4}$$

где TP и TF используются для обозначения правильных положительных и негативных ответов, а FP и FN — неправильных положительных и негативных, соответственно.

Так как классификация у нас не бинарная, метрики соответственно видоизменяются: вычисляется метрика отдельно по классу, как если бы данный класс был «положительным», остальные — отрицательным, и полученные значения усредняются по классам.

Также нас будет интересовать точность и полнота отдельно для классов «-1» и «1» .

ГЛАВА 5
ОБУЧЕНИЕ И ТЕСТИРОВАНИЕ МОДЕЛЕЙ

5.1 Предобработка данных

Все данные фильтровались по времени суток с целью избежания «особенных» периодов: открытия, закрытия и т. д..

Далее формировались признаки по одному из рассмотренных подходов и полученные данные нормализовались по формуле:

$$z_i = \frac{x_i - \overline{x}}{\sigma}$$

Стандартное отклонение и среднее для нормализации считались на обучающей выборке.

Целевая переменная преобразовывалась в соответствии с описанием в разделе 4.1.

5.2 Обучение и тестирование

Построение и оптимизация моделей проводились с помощью фреймворка Keras на основе библиотеки Tensorflow.

Размер батча при обучении задавался 128 для всех сетей, кроме CNN — здесь параметр был равен 64. Для оптимизации использовался оптимизатор Adam. Функция потерь — категориальная перекрестная энтропия. Количество эпох для всех сетей было равно 30 (или 60, если не происходило переобучение), но с заданным правилом остановки обучения заранее при изменении функции потерь менее чем на 0.001 в течении 10 эпох.

Так как выборка несбалансированная, при обучении указывались веса классов.

Далее представлены графики кривых обучения и валидации для функции потерь и $f1$–функции.

5.2.1 SimpleLSTM

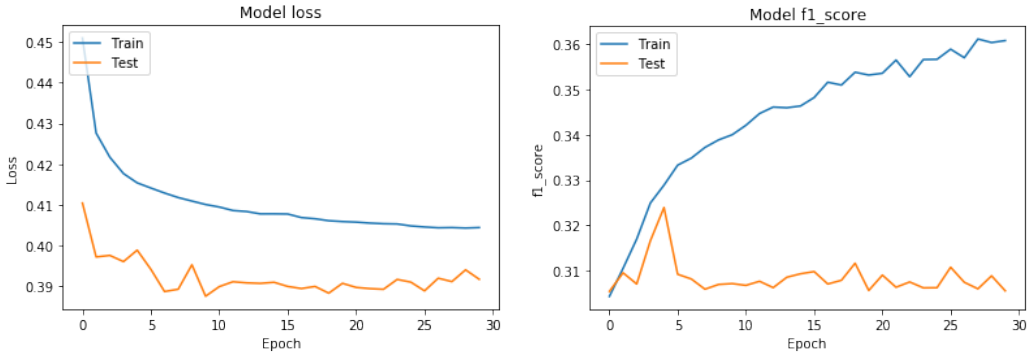


Рисунок 5.1: SimpleLSTM. Raw values

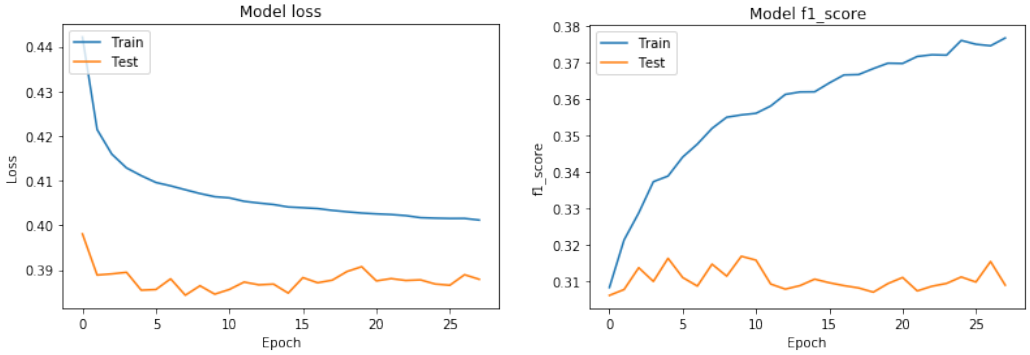


Рисунок 5.2: SimpleLSTM. Stationary features

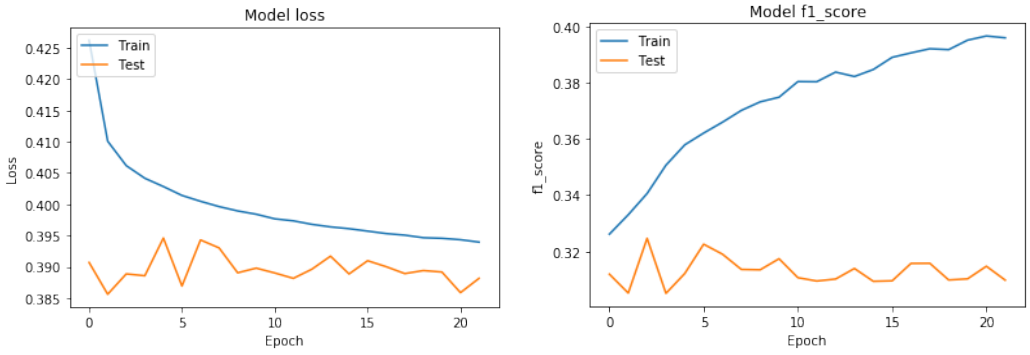


Рисунок 5.3: SimpleLSTM. Delta features

5.2.2 CNN

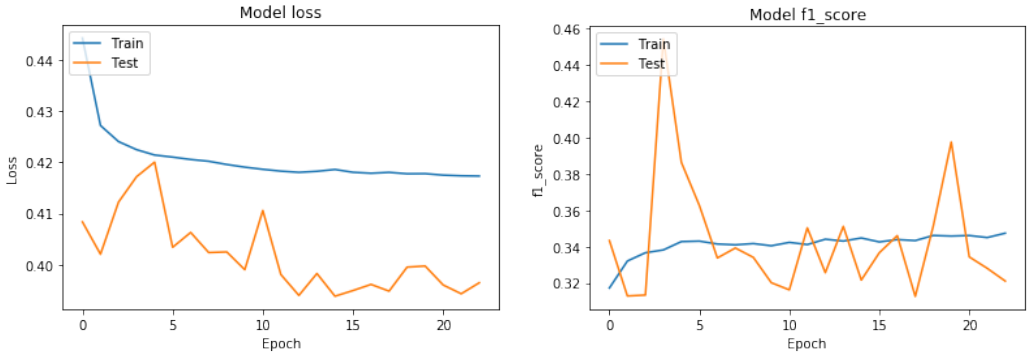


Рисунок 5.4: CNN. Raw values

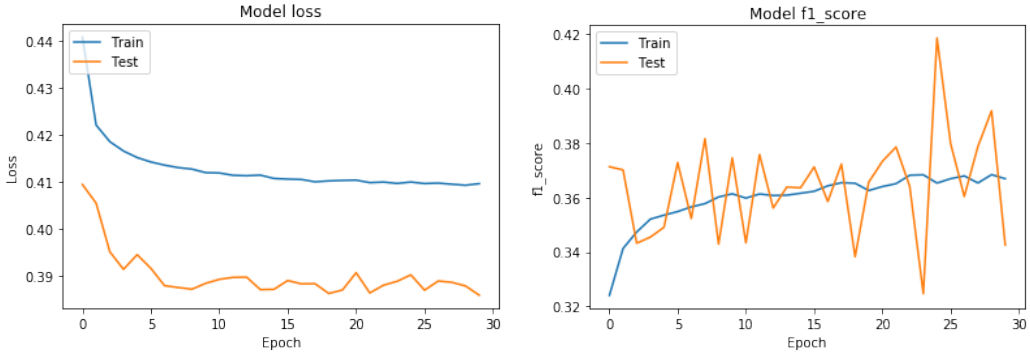


Рисунок 5.5: CNN. Stationary features

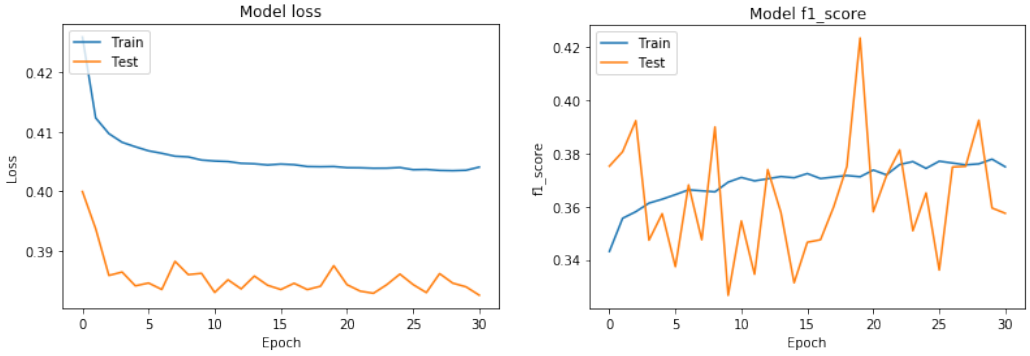


Рисунок 5.6: CNN. Delta features

5.2.3 ConvLSTM

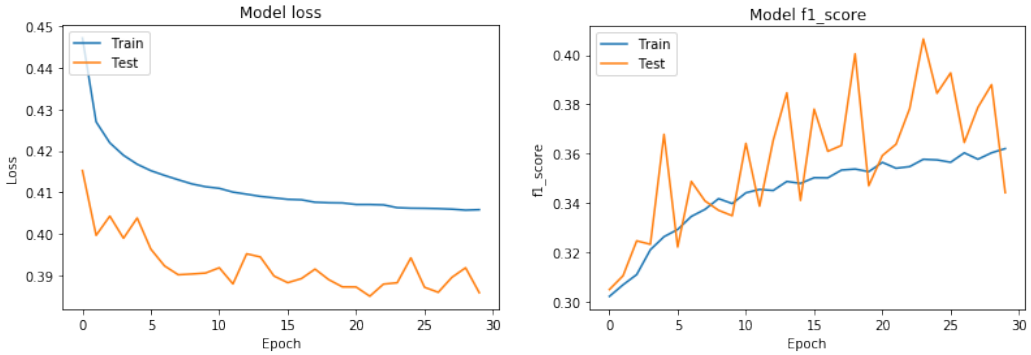


Рисунок 5.7: ConvLSTM. Raw values

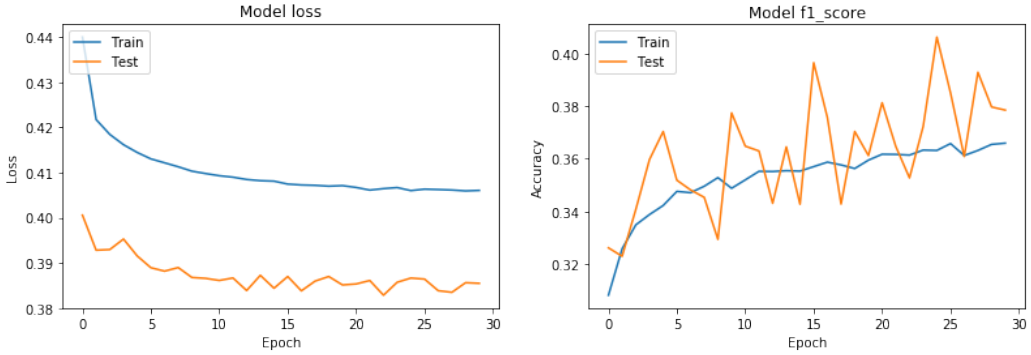


Рисунок 5.8: ConvLSTM. Stationary features

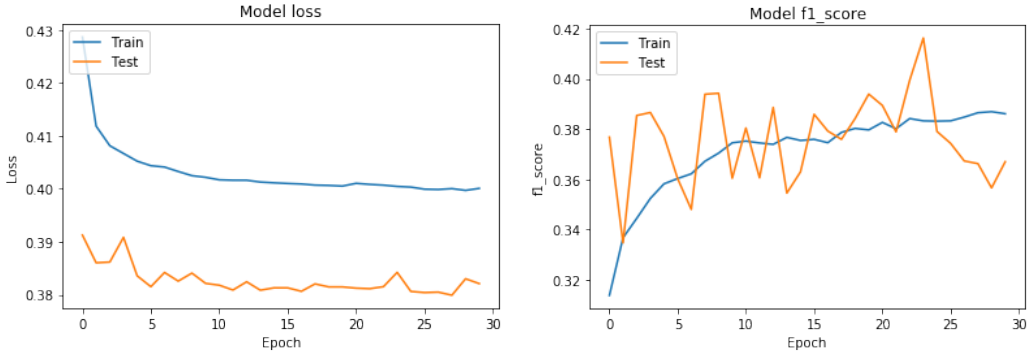


Рисунок 5.9: ConvLSTM. Delta features

5.2.4 DeepLOB

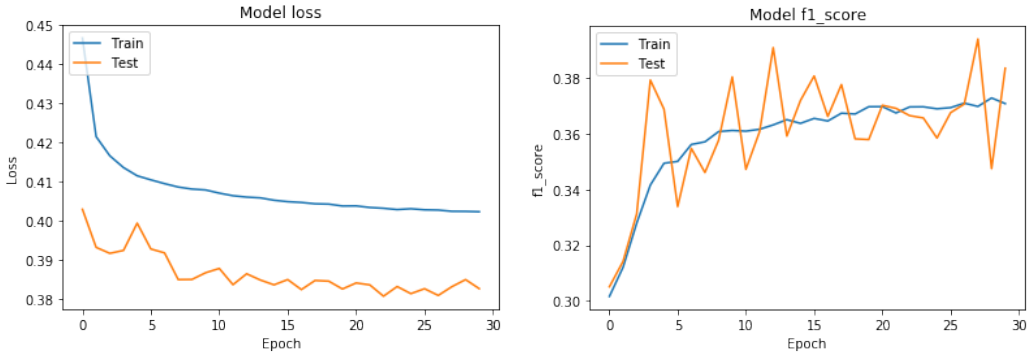


Рисунок 5.10: DeepLOB. Raw values

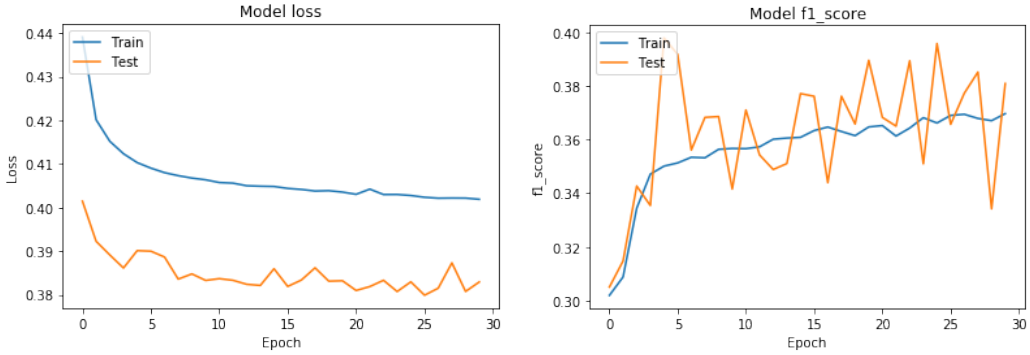


Рисунок 5.11: DeepLOB. Stationary features

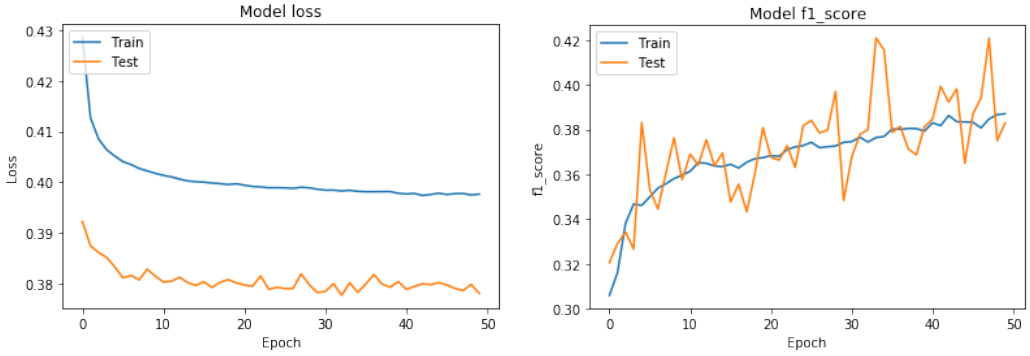


Рисунок 5.12: DeepLOB. Delta features

5.2.5 GAN–FD

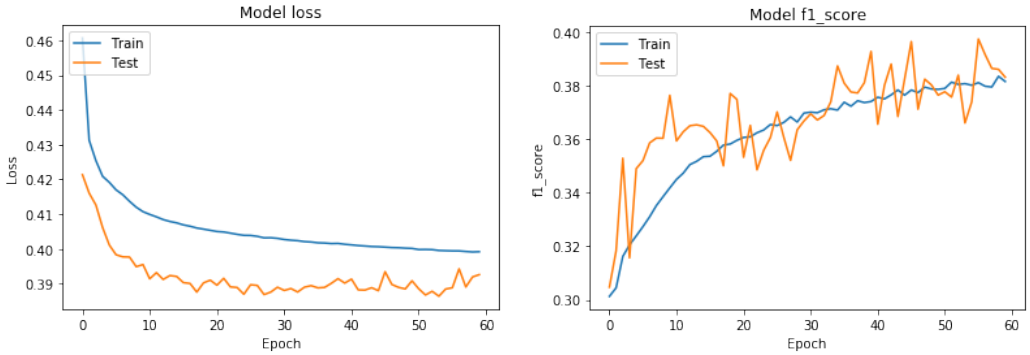


Рисунок 5.13: GAN–FD. Raw values. Generator

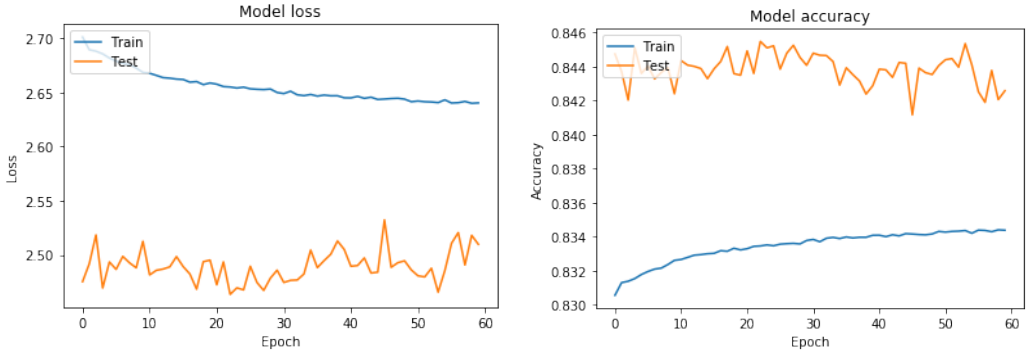


Рисунок 5.14: GAN–FD. Raw values. Discriminator

5.3 Сравнение результатов моделей на разных наборах признаков

В таблице 5.1 представлены значения метрик для различных архитектур на тестовой выборке.

Таблица 5.1: Метрики качества моделей

	Precision	**Recall**	**F1**	**Precision**$_{\{-1\}}$	**Precision**$_{\{1\}}$	**Recall**$_{\{-1\}}$	**Recall**$_{\{1\}}$
Raw values							
SimpleLSTM	0.64	0.33	0.31	0.60	0.47	< 0.01	< 0.01
CNN	0.62	0.34	0.33	0.51	0.51	0.01	0.02
ConvLSTM	0.66	0.35	0.35	0.58	0.55	0.03	0.03
DeepLOB	0.64	0.38	0.40	0.51	0.53	0.09	0.07
GAN–FD	0.64	0.37	0.39	0.55	0.50	0.07	0.07
Stationary features							
SimpleLSTM	0.67	0.33	0.31	0.70	0.45	< 0.01	< 0.01
CNN	0.68	0.35	0.33	0.58	0.61	0.05	0.02
ConvLSTM	0.66	0.36	0.36	0.56	0.56	0.10	0.07
DeepLOB	0.64	0.38	0.40	0.53	0.54	0.08	0.08
Delta features							
SimpleLSTM	0.80	0.34	0.31	0.78	0.78	0.01	< 0.01
CNN	0.66	0.38	0.37	0.59	0.54	0.04	0.07
ConvLSTM	0.67	0.37	0.38	0.59	0.55	0.07	0.06
DeepLOB	0.65	0.38	0.40	0.56	0.52	0.08	0.09

5.3.1 Сравнение архитектур и оценка результатов

Для принятия решений в торговых алгоритмах как правило важна точность предсказания движения («-1» , «1» классов).
Это соответственно Precision$_{\{-1\}}$ и Precision$_{\{1\}}$. Тем не менее количество сигналов, сгенерированное моделью также значимо: Recall$_{\{-1\}}$ и Recall$_{\{1\}}$. Можно наблюдать некоторый trade–off по этим двум показателям.

Значения метрик, полученные для моделей на исследуемом наборе данных, ожидаемы и соотносятся с представленными в статьях по аналогичной задаче и методам. Однако это не позволяет их использовать непосредственно, как чать автоматической трейдинг–системы. Но по ним можно определить

потенциальных кандидатов для дальнейшей разработки алгоритмов.

Хорошим свойством исследованных алгоритмов является также то, что они редко «ошибаются» в противоположном направлении, т. е. предсказывают «–1» на примере из «1» и наоборот.

Для примера приведем матрицы ошибок для некоторых алгоритмов и наборов данных.

	-1	0	1
-1	132	25894	0
0	38	302688	31
1	0	26020	113

Таблица 5.2: Матрица ошибок. SimpleLSTM. Delta features

	-1	0	1
-1	1733	24283	10
0	1196	300264	1297
1	8	24530	1559

Таблица 5.3: Матрица ошибок. ConvLSTM. Delta features

	-1	0	1
-1	2005	23999	22
0	1583	299095	2079
1	17	23794	2322

Таблица 5.4: Матрица ошибок. DeepLOB. Delta features

Здесь мы видим, что не смотря на то, что точность модели имеют точность не более 70% (за исключением SimpleLSTM его точность 80%), основная часть ошибок «опасных» для нас ошибок, это предсказание «–1» или «1» , тогда как на самом деле «0». Так как в этом случае стратегия реагирует на сигнал и ошибка приводит к финансовым потерям. Ошибка вида «0», когда на самом деле «–1» или «1» могут рассматриваться как менее опасные (хотя

конечно же все зависит от логики алгоритма, здесь приведены рассуждения основывающиеся на практическом опыте автора).

Сеть GAN–FD, адоптированная в данной работе для данных книги заявок, не позволила получить более высокую точность по интересующим классам, но вместе с DeepLOB дает лучшие результаты по полноте. В качестве дальнейшего развития даного подхода может быть рассмотрена генерация следующего снимка книги (горизонта снимков книг).

В целом можно заметить, что гибридные сети решают задачу лучше в совокупности по Precision и Recall. Самой устойчивой моделью можно назвать ConvLSTM. Также она сравнительно недолго обучается.

5.3.2 Сравнение способов задания книги заявок

Если сравнивать наборы признаков между собой, то предложенные в данной работе Delta features дают немного лучший результат практически для всех алгоритмов.

В целом можно заметить, что как Stationary features так и Delta features позволяют достичь более высокого качества в сравнении с Raw values, при этом преобразования, которые они включают, не являются трудоемкими и могут быть имплементированы в автоматических торговых алгоритмах.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была рассмотрена задача прогнозирования цен фьючерсных контрактов по временным данным книги заявок.

Был осуществлен сбор и обработка данных. Были рассмотрены различные форматы входных данных, а также предложен новый подход к генерации признаков, который позволяет улучшить качество предсказания.

Далее были исследованы, реализованы и протестированы различные существующие архитектуры нейронных сетей, используемых для прогнозирования финансовых рядов на имеющихся данных.

Был адаптирован метод, использующий генеративно–состязательные сети для предсказания одномерных рядов с долгим горизонтом предсказания, для имеющихся данных.

Проведен сравнительный анализ моделей и интерпретация результатов. Выделены модели и подходы к заданию книги заявок — наиболее перспективные кандидаты для разработки на их базе автоматических торговых алгоритмов.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

[1] Dimitar Bogoev and Arzé Karam. Detection of algorithmic trading. *Physica A: Statistical Mechanics and its Applications*, 484(C):168–181, 2017.

[2] Svetlana Borovkova and Ioannis Tsiamas. An ensemble of lstm neural networks for high-frequency stock market classification. *Journal of Forecasting*, 38(6):600–619, 2019.

[3] Matthew Dixon. Sequence classification of the limit order book using recurrent neural networks. *Journal of Computational Science*, 24:277–286, 2018.

[4] Larry Harris. *Trading and Exchanges: Market Microstructure for Practitioners*. Oxford University Press, 2002.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[6] Alec N. Kercheval and Yuan Zhang. Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, 15(8):1315–1329, 2015.

[7] János Levendovszky and Farhad Kia. Prediction based – high frequency trading on financial time series. *Periodica Polytechnica Electrical Engineering and Computer Science*, 56(1):29–34, 2012.

[8] M. Magris, J. Kim, E. Räsänen, and J. Kanniainen. Long-range auto-correlations in limit order book markets: Inter-and cross-event analysis. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, Nov 2017.

[9] Milla Mäkinen, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Forecasting of jump arrivals in stock prices: New attention-based network architecture using limit order book data. *SSRN*, 10 2018.

[10] Adamantios Ntakaris, Martin Magris, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, 2017.

[11] Adamantios Ntakaris, Giorgio Mirone, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Feature engineering for mid-price prediction with deep learning. *IEEE Access*, 7:82390–82412, 2019.

[12] Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Temporal logistic neural bag-of-features for financial time series forecasting leveraging limit order book data. *CoRR*, abs/1901.08280, 2019.

[13] Mehreen Rehman, Gul Muhammad Khan, and Sahibzada Ali Mahmud. Foreign currency exchange rates prediction using cgp and recurrent neural network. *IERI Procedia*, 10:239 – 244, 2014. International Conference on Future Information Engineering (FIE 2014).

[14] Justin A. Sirignano. Deep learning for limit order books. *Quantitative Finance*, 0(0):1–22, 2018.

[15] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[16] Dat Thanh Tran, Alexandros Iosifidis, Juho Kanniainen, and Moncef Gabbouj. Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE transactions on neural networks and learning systems*, 30(5):1407–1418, 2018.

[17] A. Tsantekidis, N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Forecasting stock prices from the limit order book using convolutional neural networks. In *2017 IEEE 19th Conference on Business Informatics (CBI)*, volume 01, pages 7–12, July 2017.

[18] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Using deep learning for price prediction by exploiting stationary limit order book features. *CoRR*, abs/1810.09965, 2018.

[19] Guyu Hu Siqi Tang Xingyu Zhou, Zhisong Pan and Cheng Zhao. Stock market prediction on high-frequency data using generative adversarial nets. *Mathematical Problems in Engineering*, 2018, April 2018.

[20] Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. Practical deep reinforcement learning approach for stock trading, 2018.

[21] Kang Zhang, Guoqiang Zhong, Junyu Dong, Shengke Wang, and Yong Wang. Stock market prediction based on generative adversarial network. *Procedia Computer Science*, 147:400 – 406, 2019. 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.

[22] Zihao Zhang, Stefan Zohren, and Stephen Roberts. Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012, 2019.