



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***  
***НА ТЕМУ:***

**«Метод прогнозирования временных рядов на фондовом  
рынке»**

Студент группы **ИУ7-83Б**

\_\_\_\_\_  
(Подпись, дата)

**Ахметов К.Р.**

\_\_\_\_\_  
(И.О. Фамилия)

Руководитель ВКР

\_\_\_\_\_  
(Подпись, дата)

**Филиппов М.В.**

\_\_\_\_\_  
(И.О. Фамилия)

Нормоконтролер

\_\_\_\_\_  
(Подпись, дата)

\_\_\_\_\_  
(И.О. Фамилия)

**2022 г.**

## РЕФЕРАТ

Расчетно-пояснительная записка 72 с., 21 рис., 3 табл., 30 ист., 1 прил.

Объектом разработки является комбинированный метод прогнозирования временных рядов на фондовой бирже.

Цель работы - исследовать существующие решения, разработать на их основе комбинированный метод прогнозирования временных рядов на фондовой бирже.

Для достижения поставленной цели потребуется:

- Описать предметную область и обозначить проблему;
- Провести анализ существующих методов и средств анализа временных рядов, отобрать наиболее удовлетворяющие параметрам эффективности и программной реализуемости;
- Разработка выбранных методов;
- Разработка комбинирования реализованных методов;

# СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ</b>	<b>9</b>
<b>1 Аналитический раздел</b>	<b>10</b>
1.1 Анализ предметной области . . . . .	10
1.1.1 Постановка задачи . . . . .	10
1.1.2 Основные понятия . . . . .	11
1.2 Анализ существующих решений . . . . .	14
1.2.1 Оценка эффективности классических методов анализа фондового рынка	14
1.2.2 Методы на основе теории детерминированного хаоса . . . . .	16
1.2.3 Инструменты на основе разработок в области ИИ . . . . .	17
1.2.4 Резонансные системы . . . . .	19
1.2.5 Алгоритм адаптивной фильтрации . . . . .	20
1.2.6 Генетический алгоритм . . . . .	22
1.2.7 Выводы на основе анализа существующих решений . . . . .	23
1.3 Анализ существующих моделей прогнозирования временных рядов . . . . .	24
1.3.1 Классификация моделей . . . . .	24
1.3.2 Концепция комбинированной модели . . . . .	25
1.4 Выводы из аналитического раздела . . . . .	25
<b>2 Конструкторский раздел</b>	<b>26</b>
2.1 Авторегрессионные модели . . . . .	26
2.1.1 ARIMA . . . . .	26
2.2 Нейросетевые модели . . . . .	30
2.2.1 Глубинное обучение на последовательности данных . . . . .	30
2.2.2 Механизм внимания . . . . .	33
2.3 Предложенная модель . . . . .	34
2.4 IDEF0-диаграмма . . . . .	37
2.5 Выводы из конструкторского раздела . . . . .	37
<b>3 Технологический раздел</b>	<b>39</b>
3.1 Выбор и обоснование языка программирования, среды разработки и задействованных библиотек . . . . .	39
3.2 Задействованные данные . . . . .	40

3.3	Структура разработанного программного комплекса . . . . .	42
3.4	Выводы из технологического раздела . . . . .	43
<b>4</b>	<b>Исследовательский раздел</b>	<b>44</b>
4.1	Эффективность предсказаний . . . . .	44
4.2	Сравнение комбинаций разработанных моделей . . . . .	49
4.3	Выводы из исследовательского раздела . . . . .	50
	<b>ЗАКЛЮЧЕНИЕ</b>	<b>51</b>
	<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>53</b>
	<b>ПРИЛОЖЕНИЕ А</b>	<b>58</b>

## ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

- 1) ARIMA (от англ. Autoregressive integrated moving average) — интегрированная модель авторегрессии — скользящего среднего — модель и методология анализа временных рядов.
- 2) CNN (от англ. Convolutional neural network) — сверточная нейронная сеть.
- 3) ACNN (от англ. Attention Convolutional Neural Network) — модель сверточной нейронной сети на основе механизма внимания.
- 4) RNN (от англ. Recurrent neural network) — рекуррентная нейронная сеть.
- 5) LSTM (от англ. long-short Term Memory) — разновидность архитектуры рекуррентных нейронных сетей, способная запоминать значения как на короткие, так и на длинные промежутки времени.
- 6) BiLSTM (от англ. Bidirectional long-short Term Memory) — это модель обработки последовательности, состоящая из двух LSTM: одна принимает входные данные в прямом направлении, а другая — в обратном.
- 7) Seq2Seq (от англ. Sequence to sequence) — семейство подходов машинного обучения, состоящее из двух рекуррентных сетей: кодировщика и декодировщика.
- 8) ADF тест (от англ. Augmented Dickey–Fuller test) — методика анализа временных рядов для проверки на стационарность.
- 9) FFNN (от англ. feed-forward neural network) — нейронная сеть, в которой соединения между узлами не образуют цикл.
- 10) HMM (от англ. Hidden Markov Model) — модель, имитирующая работу процесса, похожего на марковский процесс с неизвестными параметрами, и задачей ставится разгадывание неизвестных параметров на основе наблюдаемых.

## ВВЕДЕНИЕ

Задача прогнозирования временных рядов на фондовом рынке была и остается актуальной, поскольку предсказание является необходимым элементом любой инвестиционной деятельности. Для эффективного анализа рынка требуются соответствующие современным требованиям методы.

Цель работы — исследовать существующие решения, разработать на их основе комбинированный метод прогнозирования временных рядов на фондовой бирже.

Для достижения поставленной цели потребуется:

- Описать основные понятия предметной области и обозначить проблему;
- Провести анализ существующих методов и средств прогнозирования временных рядов;
- Отобрать наиболее удовлетворяющие параметрам эффективности методы.
- Разработка выбранных методов;
- Разработка комбинирования реализованных методов;

## **1 Аналитический раздел**

В данном разделе рассматривается задача прогнозирования временных рядов на фондовом рынке. Происходит постановка задачи, описываются основные понятия предметной области. Дается анализ существующих методов и средств прогнозирования временных рядов с дальнейшим отбором наиболее эффективных для комбинирования. Формулируется концепция комбинированного метода на уровне идеи, происходит формализация задачи.

### **1.1 Анализ предметной области**

#### **1.1.1 Постановка задачи**

С того момента как Россия вступила на рыночный путь развития – рынок ценных бумаг занял лидирующее место среди институтов новой экономики. В наше время наблюдается активный рост количества частных инвесторов на Московской бирже [1]. Причина данного явления кроется в главном свойстве ценной бумаги — это способность приносить доход. Также особую привлекательность добавляет тот факт, что грамотное распределение денежных средств на фондовом рынке потенциально обладает большей прибыльностью, нежели банковский вклад.

Совершенно очевидно, что различного рода анализ финансовых временных рядов с целью прогнозирования непосредственным образом превращается в прибыль. Тем не менее важно понимать, что бурный рост финансовых рынков сопровождается соответствующим ростом порога вхождения. Подтверждением этого служит статистика о том, что большая часть инвесторов, совершающих операции на фондовом рынке имеет достаточно низкий уровень общей прибыльности операций, некоторые же терпят значительные убытки [2][3]. Одной из возможных причин этому служит необходимость конкурирования рядового игрока с нарастающей автоматизацией отрасли биржевой торговли с эффективным инструментарием [5].

Финансовые боты задействованы в основном в сверхскоростных сдел-

ках, совершаемых за долю секунды, именуемыми скальпингом. Исходя из этого факта и заключения статьи [9], наиболее выигрышным решением сложившейся ситуации представляется использование инструментов прогнозирования на более длительном отрезке времени.

Также стоит отметить, что прогнозирование котировок акций — комплексная и сложная задача, представляющая собой анализ огромного количества факторов, которые невозможно учесть в полном объеме. Ситуация дополнительно усугубляется появлением человеческого фактора, неизбежно ведущим к стохастичности свойств и неопределенности экономической системы. Причиной этому служит крайне сложная структура модели поведения человека, которая зачастую может быть иррациональной. [6] Определяет ее, как правило доступные информационные ресурсы. [7] При чем вовсе не обязательно данные ресурсы соответствует действительности. [8]

На основании озвученной проблемы возникает потребность в классификации существующих методов прогнозирования временных рядов на фондовом рынке, которые станут основой для разработки комбинированного метода.

### **1.1.2 Основные понятия**

#### **Классические методы анализа фондового рынка**

Существует три основных метода анализа фондового рынка: фундаментальный анализ, технический анализ и интуитивный подход к анализу. [4]

Фундаментальный анализ — анализ финансовой деятельности компании с целью адекватной оценки ценных бумаг, при этом особое внимание уделяется будущим доходам компании, ожидаемым дивидендам и будущим процентным ставкам, а так же оценка риска деятельности компании. [4]

Технический анализ — основан на следующей гипотезе, утверждающей, что рыночные цены учитывают все знания, желания и действия всех участников рынка, отражая их в своей динамике. В результате и цена, и объем включают в себя каждую сделку, совершенную многотысячной армией трейдеров. Исходя из этой гипотезы существует значительное количество индикаторов для



определения моментов покупки и продажи ценных бумаг. [4]

Фундаментальный и технический анализ – оба направления по сути основываются на статистических данных о рынках, но находятся на противоположенных сторонах. Первый дает возможность разработки долгосрочной стратегии, а второй — только краткосрочной.

Интуитивный (психологический) подход к анализу — наименее рациональный среди перечисленных, зачастую не причисляется к классическим методам. Как правило, не приводит к долговременному успеху ввиду его нестабильности. Применение оправдано только при наличии обширного опыта, в сочетании с необходимостью быстрой реакции на стремительно изменяющиеся тенденции финансового рынка.

На практике рекомендуется использовать элементы всех 3 видов анализа. Это не конкурирующие, а взаимодополняющие друг друга инструменты.

### **Финансовые временные ряды**

Временной ряд — это определенный признак, значение которого отслеживается через постоянные временные интервалы. Измерения признака происходят во времени, и между разными измерениями проходит одинаковое количество времени, это является ключевой особенностью временных рядов, поскольку в случае, когда промежутки между отчетами различные, то этот процесс является случайным, следовательно, он неупорядочен во времени и целевая переменная не зависит от исторических данных. Методики анализа случайных процессов и временных рядов значительно отличаются, следовательно, важно обращать внимание на данную особенность.[21]

Задачу прогнозирования финансовых временных рядов можно определить как:

- *Классификацию*. Предсказание качественного отклика для некоторого наблюдения можно отнести к определенной категории, или классу;
- *Регрессию*. Исследование влияния одной группы непрерывных случайных величин на другую.

В рамках фондового рынка временные ряды, как правило, основываются на ценах или их динамике, такие временные ряды называют финансовыми.

При моделировании финансовых временных рядов часто можно столкнуться со следующими проблемами:

- *Автокорреляция*. Предсказание качественного отклика для некоторого наблюдения можно отнести к определенной категории, или классу;
- *Нестационарность*. Цены акций зависят от большого количества факторов извне, характеризующихся компонентами тренда, сезонности, циклов, всплесков, которые в свою очередь, зависят от времени;
- *Гетероскедстичность*. Участки с низкой волатильностью часто сменяются участками с высокой волатильностью;
- *Наличие в данных выбросов*. Выбросы представляют собой точки, которые значительно отличаются от остальных наблюдений временного ряда.

## **1.2 Анализ существующих решений**

### **1.2.1 Оценка эффективности классических методов анализа фондового рынка**

#### **Фундаментальный анализ**

Методы фундаментального анализа вызывают разные оценки у специалистов [22]. С одной стороны, фундаментальный анализ работает с реальными причинами движения цен — объективными факторами и причинноследственными закономерностями, известными в экономике и финансовой науке. Обычно он проводится на нескольких уровнях — от макроэкономического до уровня финансового состояния самого эмитента, благодаря чему формируется полноценное исследование объекта инвестирования и общей ситуации принятия решений. Поэтому прогнозирование на его основе представляется достаточно хорошо проработанным и экономически обоснованным.

С другой стороны применение фундаментального анализа содержит в себе и некоторые проблемы. Первая из них связана с гипотезой эффективного рынка. Основная предпосылка фундаментального анализа строится на том, что рыночная цена акции будет приближаться к своему «справедливому» значению ввиду действия гипотезы о рыночной эффективности. При этом забывается, что если бы данная гипотеза выполнялась, то наблюдаемая рыночная неэффективность с точки зрения базовой логики вообще не могла бы возникнуть.

В целом, методы фундаментального анализа являются достаточно сложными, грамотное их использование доступно только профессиональным инвесторам и аналитикам, долгое время работающим на рынке. Важно также помнить о том, что возможным недостатком фундаментального анализа является его ориентация на долгосрочные вложения.

#### **Технический анализ**

Методы технического анализа могут применяться для совершения инвестиционных операций на современной бирже, однако эффективность данного

метода находится под большим вопросом. [22] Причиной сложившейся ситуации является ряд аргументов:

- Технический анализ основан на использовании единственного вида информации — цен прогнозируемого рыночного актива, поэтому с его помощью невозможно оценить влияние иной информации;
- Традиционные методы технического анализа являются чрезмерно упрощенными и не способны отражать сложные зависимости; присутствующие в ценовой динамике рыночных активов;
- В случае использования традиционных методов технического анализа динамическая адаптация к происходящим на рынке изменениям отсутствует, что делает невозможным учет изменений, происходящих после настройки алгоритмов прогностической модели.

### **Вывод об эффективности классических методов**

Основной вывод проведенного анализа, основанного на изучении практики применения данных инструментов следующий – данные подходы имеют низкие прогностические возможности, явно недостаточные для обеспечения приемлемого уровня доходности, стабильной в долгосрочной перспективе.

Среди наиболее значительных недостатков можно выделить следующие:

- Задействование чрезмерно малого количества факторов;
- Фактически полное отсутствие аппарата преобразования проявления факторов влияния в прогнозные значения динамики курсовых стоимостей акций;
- Полная неизменность или крайне слабая изменяемость во времени зависимостей, на основе которых осуществляется расчет прогнозных значений.

### **1.2.2 Методы на основе теории детерминированного хаоса**

В работе [23] на основе теории хаоса представлена непараметрическая модель для прогнозирования будущего поведения валютных курсов. Теория хаоса предполагает, что поведение финансовых рынков хаотично и обеспечивает основу для учета динамики нелинейных систем, которые могут выявлять скрытые шаблоны и тенденции в финансовых данных, и которые, в свою очередь, не могут быть получены обычными статистическими методами. Предложенная двухступенчатая модель сочетает в себе фазовое пространство, реконструированное из одномерных хаотических временных рядов в комбинации с векторной регрессией.

Однако следует отметить, что в реальности биржевые рынки обладают свойством самоорганизации и памятью о предыдущих событиях, то есть не могут быть описаны в полной мере теорией хаоса.

Данное направление является достаточно новым и говорить о его эффективности пока, возможно, несколько преждевременно. На данный же момент, анализ специализированной литературы показал [10], что практически все публикации, в которых приводится описание прогностических инструментов на основе теории детерминированного хаоса, относятся к периодической литературе, и многие из них имеют явно рекламный характер. Ни в одном источнике не было обнаружено подробного описания применения инструментов данного класса в условиях реального фондового рынка, с указанием результатов, которые можно было бы считать достоверными и устойчивыми во времени. Несмотря на то, что в публикациях, посвященных данному направлению, иногда можно встретить описания положительных результатов применения данного подхода в прогнозировании динамики курсовых стоимостей акций, они имеют явно предвзятый характер.

Отзывы же практиков говорят прямо об обратном. Мнения непосредственных инвесторов фондового рынка сводятся скорее к тому, что эффективность

подобных инструментов весьма сомнительна, и вряд ли можно утверждать, что она превышает эффективность прогностических инструментов, созданных на основе традиционных подходов. Поэтому и распространенность инструментов на основе теории детерминированного хаоса среди инвесторов весьма мала.

### **1.2.3 Инструменты на основе разработок в области ИИ**

Более длительную историю, по сравнению с теорией хаоса, имеет подход, основанный на применении для прогнозирования динамики курсовых стоимостей акций фондового рынка разработок в области ИИ.

Искусственный интеллект — раздел информатики, изучающий возможность воспроизведения рассуждений и действий, аналогичных мышлению и действиям человека, с помощью электронных вычислительных систем и иных искусственных устройств. Главной задачей разработки систем искусственного интеллекта является построение искусственной системы, например — компьютерной программы, обладающей уровнем эффективности решения неформализованных задач, сравнимым с человеческим или превосходящим его.

Исторически выделилось два основных подхода к построению систем ИИ: логический и нейрокибернетический.

1. Логический подход — математическое моделирование сознательной деятельности человека, то есть моделирование высокоуровневых процессов мышления, основанных на представлении и использовании знаний. Методология данного подхода - выявление и применение в интеллектуальных системах различных логических и эмпирических приемов (эвристик), которые человек использует для решения каких-либо задач. С развитием в начале 80-х годов концепции экспертных систем это направление вылилось в научно-технологическое направление информатики «инженерия знаний», занимающееся созданием так называемых «систем, основанных на знаниях». Алгоритмы решения задач, используемые в таких системах, закладываются экспертами, в виде аксиоматических правил.

2. Нейрокибернетический (нейросетевой) подход — основан на матема-

тическом моделировании функционирования низкоуровневых структур мозга — нейронов. Его методология — построение самоорганизующихся систем, состоящих из множества элементов, функционально подобных нейронам головного мозга. Нейрокибернетическое направление относят к моделированию мышления на подсознательном уровне (моделирование интуиции, творческого воображения, инсайта).

Сферы применения разработок в области ИИ крайне разнообразны: медицина, геология, военная техника, экономика и др. Такие разработки используются для решения задач предсказания, диагностики, планирования, управления. Нейронные сети способны решать задачи в условиях неполноты, искаженности, зашумленности информации, а также при отсутствии алгоритма решения задач.

Задача прогнозирования динамики курсовых стоимостей акций фондового рынка относится к неформализованным, поэтому для ее решения используются разработки в области нейрокибернетического подхода — нейронные сети.

На сегодняшний день такие нейросетевые пакеты, получили достаточно широкое распространение на фондовом рынке, особенно в странах Запада. Среди наиболее известных программных продуктов можно назвать: NeuroShell Day Trader, Neuro Builder, Brain Maker, NeuroScalp.

Несмотря на распространенность, при более тщательном анализе можно утверждать, что на практике нейронные сети, по крайней мере реализованные в стандартных специализированных программных пакетах, имеют не самую выдающуюся прогностическую эффективность.[11] Причиной этого явления послужил ряд важных недостатков:

- Не обеспечивается непрерывное преобразование («переобучение») модели на основе новой входящей информации, в том числе на основе опыта решения задач, поставленных перед инструментом, в процессе его функционирования;
- Нейросетевая структура не отражает преобразование информации в

психике человека, поэтому не позволяет имитировать совершение инвесторами массовых фондовых операций на основе известной входной информации. Иными словами, один из наиболее значительных факторов не может быть реализован в полной мере.

#### **1.2.4 Резонансные системы**

На динамику курсовых стоимостей акций фондового рынка влияет только та информация, которая используется инвесторами при принятии решений о совершении фондовых операций, само же влияние данной информации на совершение инвесторами фондовых операций определяется механизмами преобразования мозгом человека входной информации в принимаемые решения.

Основой резонансной системы является модель преобразования информации, относящейся к фондовому рынку в совершаемые инвесторами операции, так как она показывает, какие именно факторы инвесторы используют и как они преобразуются в решения о совершении фондовых операций.

Представляет собой систему механизмов преобразования мозгом входной информации в принимаемые решения, то есть систему механизмов мышления, которые моделируются не на нейронном уровне, а на более глубинном материальном уровне — волновом.[20]

Возможность представления резонансной модели в математическом виде обеспечило потенциал к применению данного подхода для построения прогностических инструментов, ориентированных на определение будущих курсовых стоимостей акций фондового рынка. Сама концепция предполагает, что сущность процесса формирования котировок заключается в ответе на следующие вопросы:

- Какие факторы (сознательные-неосознаваемые) влияют на принятие инвестиционных решений;
- Как у инвесторов формируется опыт совершения фондовых операций (опыт представляется в виде набора закономерностей совершения фондовых операций при проявлении влияющих факторов);



- Как в информационных потоках фондового рынка инвесторами сознательно неосознаваемо идентифицируются факторы влияния;
- Как проявление тех или иных факторов, обрабатываясь информацией прошлого опыта, результирует в совершении инвесторами тех или иных фондовых операций;
- Как оперировать общими тенденциями данного информационного потока, который образуется действиями большого количества инвесторов.

Основная проблематика использования резонансных систем на фондовом рынке заключается в том, что в доступных источниках в открытом доступе она описана, к сожалению, только на уровне общей архитектуры.

### **1.2.5 Алгоритм адаптивной фильтрации**

Адаптивный фильтр — это инструмент, способный к самообучению для достижения заданного уровня соответствия выходных данных реальному положению вещей. Под адаптивностью понимается возможность реагировать на изменения входных данных в режиме реального времени для достижения более высокой производительности. На практике, адаптивные алгоритмы реализуются двумя классическими методами — методом градиента и наименьших квадратов.

Данный алгоритм широко применяется в радиоэлектронике в качестве системы цифровой обработки данных. Однако, с публикацией следующего исследования. [12] область применения алгоритма адаптивной фильтрации распространилась, и на предсказание поведения активов, торгуемых на фондовом рынке.

На рисунке 1 представлена универсальная схема применения адаптивного фильтра в предиктивном фреймворке, где  $k$  — это номер итерации,  $x(k)$  — входящий сигнал,  $y(k)$  — вывод адаптивного фильтра, являющийся прогнозом желаемого ответа  $d(k)$ , а  $e(k)$  — сигнал ошибки, определяемый как разница между желаемым ответом и выводом фильтра, то есть  $e(k) = d(k) - y(k)$ .

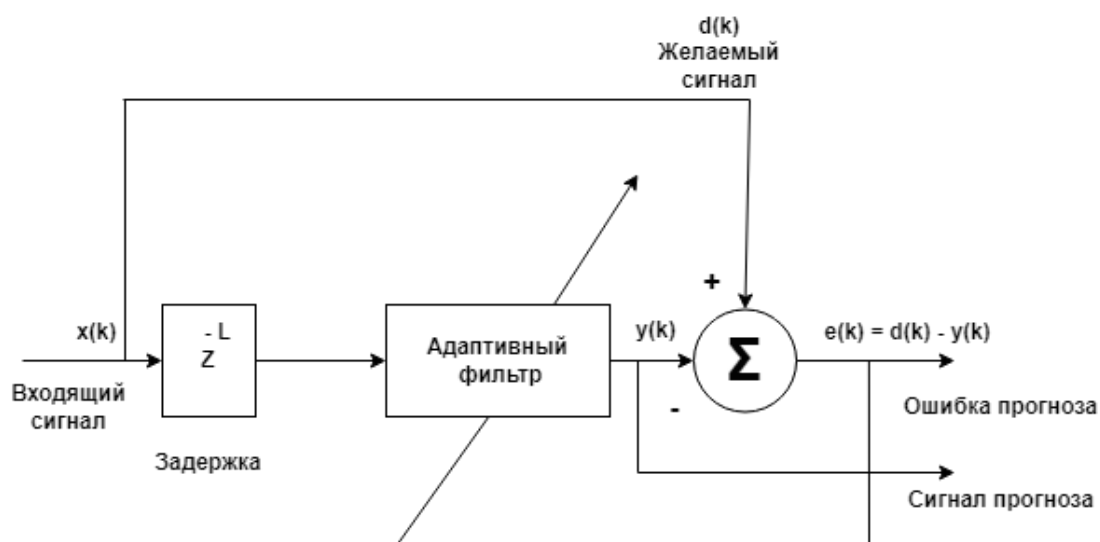


Рисунок 1 – Блок-диаграмма адаптивного фильтра для предсказания сигналов

Не углубляясь в дальнейшие детали реализации необходимо оценить эффективность данного алгоритма в качестве инструмента для предсказания цен акций на реальных биржах. По заявлениям авторов наиболее оптимальным окном предсказания является 15-20 дней, а средняя прибыль составляет около 7% от объёма инвестиций. Проблематика состоит в том, что статистика для других торговых дней не приведена с формулировкой, что результаты были схожи. Вывод исследования вызывает подозрения о манипулирование путем сокрытия части информации — в статье явно не хватает примеров, где фильтр ошибся, и соотношение ошибок и успехов на длительном периоде.

### 1.2.6 Генетический алгоритм

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём случайного подбора, комбинирования и вариации искомых параметров с использованием механизмов, аналогичных естественному отбору в природе. Одним из наиболее важных преимуществ генетических алгоритмов является отсутствие необходимости информации о поведении функции и незначительное влияние возможных разрывов на процессы оптимизации. Также как и в случае нейронных сетей, происходит уход от необходимости анализа причинно-следственных связей, путем построения «итогового» образа — целевой функции.

Основное преимущество концепции генетического алгоритма — это скрещивание, также именуемое комбинированием. В общих чертах идея генетического алгоритма в рамках поставленной задачи сводится к перебору и отбору оптимальной комбинации. Алгоритм делится на следующие этапы:

- Скрещивание;
- Селекция (отбор);
- Формирования нового поколения.

Вышеперечисленные этапы повторяются до тех пор, пока не будет получен удовлетворительный результат или произойдет одно из ниже перечисленных условий:

- Количество поколений достигнет заранее выбранного максимума;
- Исчерпано время на мутацию.

Собственно говоря отсюда и вытекает основной недостаток — необходимая комбинация для приемлемого результата требует огромных вычислительных мощностей, причём присутствует большая вероятность ее ненахождения. С очень сильной натяжкой целевую функцию можно сравнить со слоем входных нейронов и с ожиданием максимума как аналога максимизации сигнала нейрона выходного слоя. Хотя правильнее было бы говорить, что генетические

алгоритмы используются для повышения эффективности обучения нейронных сетей, но все же не могут рассматриваться как конкуренция нейронным сетям.

### **1.2.7 Выводы на основе анализа существующих решений**

На основе рассмотренных методов можно сделать заключение, о том, что наиболее оптимальным и подходящим решением задачи прогнозирования временных рядов на фондовом рынке является использование нейронных сетей, в сочетании с дополнительными алгоритмами, призванными улучшить их конечную точность. Среди возможных кандидатов наилучшими являются следующие:

- Математические модели для анализа прогнозирования стационарных временных рядов;
- Генетические алгоритмы для повышения эффективности обучения нейронных сетей;
- Сочетание нескольких различных разновидностей архитектуры нейронных сетей.

В рамках данной работы для разработки комбинированного метода будут задействованы все вышеперечисленные варианты, за исключением генетических алгоритмов, так как на основе предварительного анализа нельзя однозначно утверждать, что они позволяют добиться улучшения прогностической эффективности метода.[13, 14]

### 1.3 Анализ существующих моделей прогнозирования временных рядов

#### 1.3.1 Классификация моделей

Модели временных рядов — математические модели прогнозирования, которые стремятся найти зависимость будущего значения от прошлого внутри самого процесса и на этой зависимости вычислить прогноз. Эти модели универсальны для различных предметных областей, то есть их общий вид не меняется в зависимости от природы временного ряда. Перед началом обзора моделей стоит отметить тот факт, что названия моделей и соответствующих им методов зачастую совпадают, также принято использовать английские аббревиатуры [24]. Сами методы прогнозирования как правило бывают двух видов:

- *Интуитивные методы*. Имеют дело с индивидуальными суждениями экспертов. Используются и применяются тогда, когда объект прогнозирования либо слишком прост, либо, напротив, настолько сложен, что аналитически учесть влияние внешних факторов невозможно;
- *Формализованные методы*. Имеют дело с математическими моделями прогнозирования, то есть определяют такую зависимость, которая позволила бы вычислить будущее значение процесса. В следующей работе [25] модели разделяются на статистические и структурные.

**В статистических моделях** функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами задана аналитически. К статистическим моделям относятся следующие группы:

- Регрессионные модели;
- Авторегрессионные модели;
- Модели экспоненциального сглаживания

**В структурных моделях** функциональная зависимость между будущими и фактическими значениями временного ряда, а также внешними факторами

задана структурно. К структурным моделям относятся следующие группы:

- Нейросетевые модели;
- Модели на базе цепей Маркова;
- Модели на базе классификационно-регрессионных деревьев.

### **1.3.2 Концепция комбинированной модели**

В рамках данной работы рассматривается комбинированная модель, предполагающая возможность компенсирования недостатков одних моделей при помощи других. Раннее изложенный вывод о существующих решениях подводит к тому, что основой комбинированной модели должна выступать как и статическая модель, так и структурная. На основе данных работ [24, 25, 26, 27] есть все основания утверждать, что среди статистических моделей наиболее подходящей является autoregressive integrated moving average (ARIMA). В свою очередь среди структурных моделей, а именно нейросетевых, неплохие результаты демонстрирует long-short term memory (LSTM), являющаяся по своей сути разновидностью архитектуры рекуррентных нейронных сетей [28, 16, 15], которая в комбинации со сверточной нейронной сетью на основе механизма внимания добивается весьма достоверных результатов [29, 19, 17, 18].

## **1.4 Выводы из аналитического раздела**

Была произведена классификация существующих методов прогнозирования временных рядов на фондовом рынке. Были решены следующие задачи:

- Описаны основные понятия предметной области и обозначена проблема;
- Проведен анализ существующих методов и средств прогнозирования временных рядов;
- Отобраны наиболее удовлетворяющие параметрам эффективности методы.

## 2 Конструкторский раздел

В данном разделе углубленно описываются задействованные методы и алгоритмы. На их основе разрабатывается комбинированный метод прогнозирования временных рядов на фондовом рынке. Предложенная модель затем декомпозируется, выделяются ключевые этапы и особенности разработки, после приводится схема IDEF0.

### 2.1 Авторегрессионные модели

#### 2.1.1 ARIMA

Autoregressive integrated moving average (ARIMA) — интегрированная модель авторегрессии — скользящего среднего классическая методология и модель анализа временных рядов. Идея данного алгоритма заключается в комбинировании процессов авторегрессии, процессов интегрирования и процессов скользящего среднего. Основной недостаток модели заключается в том, что ARIMA не предполагает включение дополнительных факторов, которые могут влиять на временной ряд. Модель базируется только на информации, содержащейся в предыстории прогнозируемого временного ряда. Одним из главных преимуществ ARIMA является способность моделировать интегрированные или разностно-стационарные временные ряды.

Самые распространенные модели предсказаний для рынка акций основаны на моделях ARMA (Auto Regressive Moving Average) и ARIMA (Auto Regressive Integrated Moving Average). Модель ARMA (p,q) представляет из себя

$$s_t = a_0 + \sum_{i=1}^p a_i s_{t-i} + w_t + \sum_{i=1}^q b_i w_{t-i}, \quad (1)$$

где  $a$  и  $b$  - параметры,  $w$  - шум.

Эта модель может быть использована для стационарной последовательности  $s_{1:N}$ , что соответственно означает

$$E[s_t] = Constant \quad (2)$$

$$Cov(s_t, s_{t-k}) = Constant, \quad (3)$$

где  $t = 1, 2, \dots, N$  и  $k = 1, 2, \dots, t$ .

В случае, когда последовательность не стационарна, ARIMA (p,q,d) использует оператор разности временного ряда порядка d. В предсказаниях для рынка акций, разность первого порядка  $x_{1:N} = s_{1:N}$  (то есть  $x_k = s_k - s_{k-1}$ ) обычно считается стационарной последовательностью.

Было проведено эмпирическое исследование акций Банка Китая на китайском фондовом рынке. Цены акций фондового рынка являются публичными. Был выбран временной отрезок с 1 января 2007 по 31 марта 2022, где один день является единицей последовательности.

Тест ADF был выбран для тестирования стационарного условия временного ряда. Тестирование было проведено для первоначальной последовательности и для последовательности разности первого порядка. Результаты приведены в Таблице 1 и 2. По результатам ADF теста можно заключить, что в случаях, когда p-значение больше 0.562 или критическое значение (1%) больше -3.44, последовательность признается нестационарной, в то же время разница первого порядка последовательности — стационарной.



Разница первого и второго порядка последовательности продемонстрированы на рисунках 2 и 3.

Таблица 1 – ADF тест для изначальной последовательности

Метрика	Значение
Значение статистического теста	-2.35539
р-значение	0.154726
Кол-во использованных лагов	16
Кол-во задействованных наблюдений	3484
Критическое значение(1%)	-3.43223
Критическое значение(5%)	-2.86237
Критическое значение(10%)	-2.56721

Таблица 2 – ADF тест для разницы первого порядка

Метрика	Значение
Значение статистического теста	-14.7498
р-значение	2.49575e-27
Кол-во использованных лагов	15
Кол-во задействованных наблюдений	3484
Критическое значение(1%)	-3.43223
Критическое значение(5%)	-2.86237
Критическое значение(10%)	-2.56721

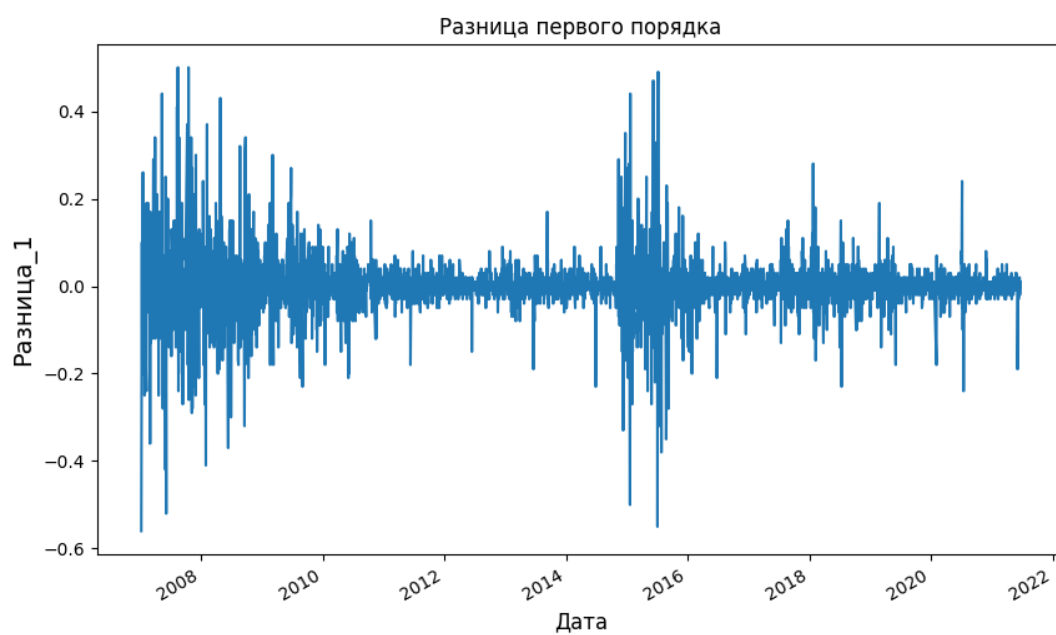


Рисунок 2 – Разница первого порядка.

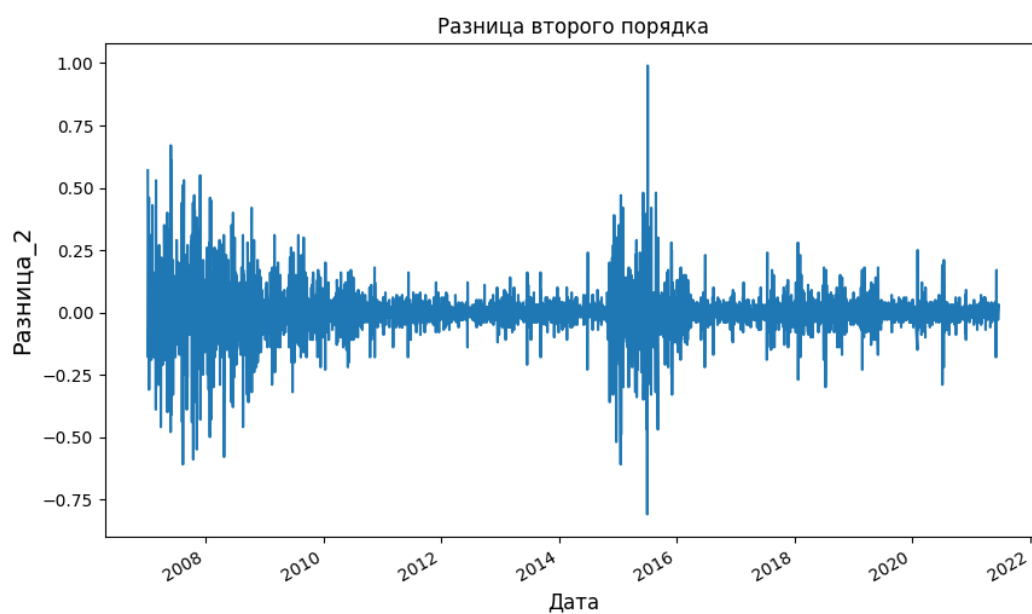


Рисунок 3 – Разница второго порядка.

## 2.2 Нейросетевые модели

### 2.2.1 Глубинное обучение на последовательности данных

Как правило, для решения задач преобразования входной последовательности временных рядов в выходную последовательность той же или иной предметной области применяются соответствующие последовательные модели. В типичной нейронной сети с прямой связью (FFNN - feed-forward neural network), вывод текущего момента  $o_t$  определяется только входом текущего момента  $i_t$ , как следствие возможности FFNN в моделирование временных рядов достаточно ограничены, что нельзя сказать о рекуррентных нейронных сетях (RNN - recurrent neural network), которые создавались с возможностью обработки длинных последовательных данных и решения задач с распределением контекста во времени. В рекуррентных нейронных сетях, используется задержка, чтобы сохранить текущее скрытое состояние последнего момента  $h_{t-1}$ , затем, скрытое состояние текущего момента  $h_t$  определяется с помощью и  $h_{t-1}$ , и  $i_t$ , что продемонстрировано на рисунке под номером 4.

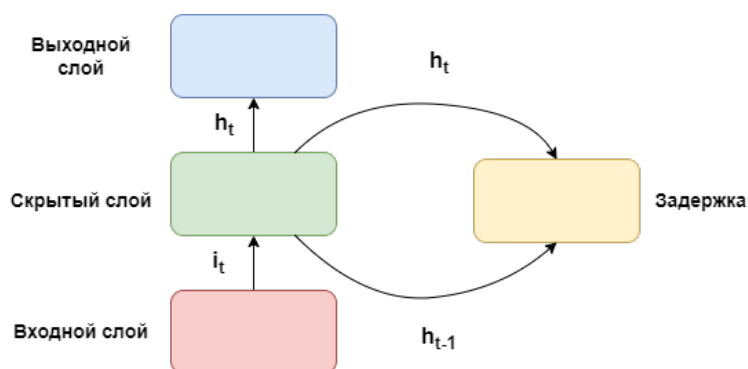


Рисунок 4 – RNN блок.

Модель обрабатывает по одному элементу последовательности на одном временном шаге. После вычисления обновлённое состояние передается на следующий шаг во времени, чтобы облегчить вычисление следующего элемента.

Тем не менее, исходя из следующей статьи [30], можно утверждать, что высока вероятность того, что RNN может легко потерять долгосрочные зависимости, поэтому для решения проблемы забывания был разработан специализи-

рованный нейрон с усложненной внутренней структурой для запоминания долгосрочного контекста, называемого ячейкой long-short Term Memory (LSTM), которая схематично отображена на рисунке 5.

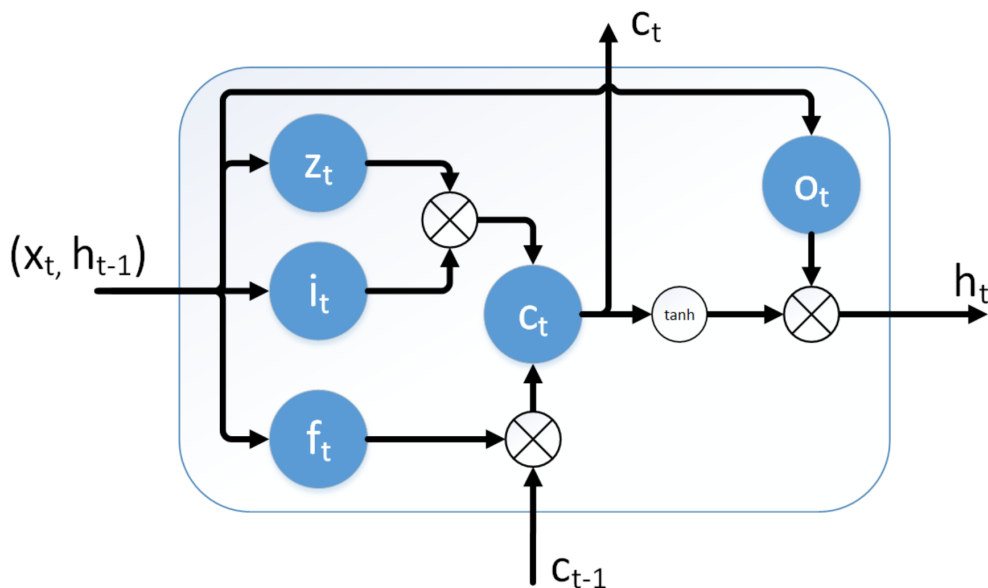


Рисунок 5 – LSTM блок.

Long-short term memory (LSTM) — достаточно распространенная архитектура рекуррентной сети, идея этой архитектуры заключается в выделение ячейки памяти, ответственной за хранение информации, полученной в предыдущие моменты времени. Для управления этой ячейкой выделяются три шлюза:

- *Шлюз памяти*. Ответственен за сохранение или забывание предыдущего состояния ячейки.
- *Входной*. Контролирует поступление данных в память;
- *Исходящий*. Контролирует распространение данных из памяти;

Функция ячейки памяти схожа с состояниями в детерминированном конечном автомате, однако в данном случае состояние «распределенное» и позволяет работать с бесконечным числом состояний.

$$\begin{aligned}
f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned} \tag{4}$$

где

$f_t$  — шлюз памяти;

$i_t$  — входной шлюз;

$o_t$  — исходящий шлюз;

LSTM сети обучаются при помощи алгоритма обратного распространения ошибки сквозь время, идея которого состоит в разворачивании графа вычислений во времени. Из LSTM сетей можно выстраивать многослойные нейронные сети, передавая выходную последовательность очередного слоя на вход следующему.

Sequence-to-sequence (seq2seq) модель использует автоматический кодировщик (модель, состоящая из двух рекуррентных сетей: кодировщика и декодировщика) для анализа последовательных данных, тем самым улучшая возможность LSTM моделей обучаться скрытой информации, не смотря на зашумленность данных. Кодировщик строит представление входной последовательности временных рядов (обычно скрытое состояние в последний момент последовательности  $h_N$ ). Далее полученное представление (последние выход и значение ячейки сети) копируются в декодировщик. По полученному представлению декодировщик пытается восстановить целевую последовательность.

Во время предсказания требуется найти наиболее вероятное продолжение последовательности временных рядов с точки зрения модели. Сделать это

напрямую невозможно, так как модель позволяет вычислять только наилучшую последовательность при фиксировании предыдущих. Компромиссным решением между жадным выбором последовательностей и полным перебором является лучевой поиск.

Лучевой поиск и алгоритм Витерби в скрытой цепи Маркова (НММ - Hidden Markov Model) оба основаны на динамическом программировании. Поиск оптимальной оценки значения текущего состояния, используя наблюдения и предыдущее состояние, называется декодированием (или интерференцией) в НММ. Вычисляя  $p(x_k|y_{1:k})$  и  $p(x_k|y_{1:N})$ , соответственно, эквивалентно алгоритму двунаправленности в НММ. Оптимальная двусторонняя оценка может быть получена через распределение  $x_k$ .

### 2.2.2 Механизм внимания

Люди, в целом, обращают внимание на заметную, выделяющуюся из общего контекста информацию. В сверточных нейронных сетях, механизм внимания — это метод, основанный на когнитивном восприятии человека. Для ввода  $X = (x_1, x_2, \dots, x_N)$ , дан вектор запроса  $q$ , индекс выбранной информации вниманием  $z = 1, 2, \dots, N$ , то распределение внимания — это

$$\alpha_i = p(z = i|X, q) = \frac{\exp(s(x_i, q))}{\sum_{j=1}^N \exp(s(x_j, q))}, \quad (5)$$

то есть,

$$\alpha_i = \text{softmax}(s(x_i, q)), \quad (6)$$

где

$$s(x_i, q) = \frac{x_i^T q}{\sqrt{d}} \quad (7)$$

Есть количество очков внимания через масштабированное скалярное произведение, а  $d$  — это измерение вводных данных. Предположим, что ввод - это пары ключ-значение  $(K, V) = [(k_1, v_1), \dots, (k_N, v_N)]$ , для данного значения  $q$ , то функция внимания — это

$$att((K, V), q) = \sum_{i=1}^N \alpha_i v_i = \sum_{i=1}^N \frac{\exp(s(k_i, q))}{\sum_j \exp(s(k_j, q))} v_i \quad (8)$$

Механизм множества внимания обычно используется через множество запросов  $Q = [q_1, \dots, q_M]$  для вычисления функции внимания:

$$att((K, V), Q) = (att((K, V), q_1) \parallel \dots \parallel att((K, V), q_M)) \quad (9)$$

Здесь, оператор  $\parallel$  обозначает оператор объединения. Это и есть, так называемый, механизм множества внимания.

Механизм внимания может быть использован для генерации различных весов на основе данных.  $Q, K, V$  получены через линейную трансформацию  $X$ , а  $W_Q, W_K, W_V$  могут быть динамически скорректированы.

$$Q = W_Q X, K = W_K X, V = W_V X \quad (10)$$

Это так называемое само-внимание. Похожим образом, вывод - это

$$h_i = att((K, V), q_i) \quad (11)$$

Используя масштабированное скалярное произведение, мы получаем

$$H = V \text{softmax}(K^T Q \frac{1}{\sqrt{d}}) \quad (12)$$

### 2.3 Предложенная модель

Реализация модели построена на комбинирование классических методов ARIMA, ACNN, BiLSTM и является гибридной моделью глубокого обучения

для предсказания курса акций. Основное отличие разработанной модели от традиционных аналогов — это интеграция модели исследования временных рядов ARIMA и нейронных сетей в нелинейной зависимости, которая позволяет объединить преимущества двух классических моделей с целью повышения точности предсказаний. Общая схема предложенной модели представлена на рисунке 6.

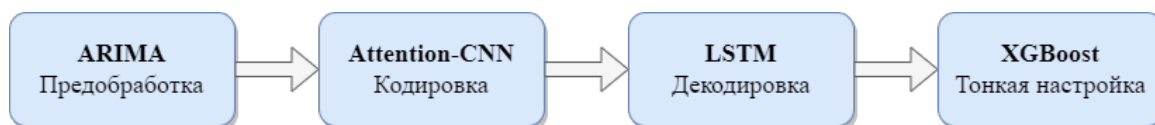


Рисунок 6 – Блок-схема разработанной комбинированной модели.

Первоначально курс акций проходит обработку через ARIMA, затем сформированная последовательность задействуется в нейронных сетях. Основная идея на этом этапе заключается в том, что изначальные данные фондового рынка помещенные в модель ARIMA стационарными взятием разностей некоторого порядка от исходного временного ряда (так называемые интегрированные или разностно-стационарные временные ряды), позволяют получить новый ряд, отображающим с большей эффективностью характеристику курса акций.

Затем архитектура глубокого обучения формируется в рамках предобучения-дообучения с применением фреймворков. Предобучающая модель представляет собой CNN-LSTM модель на основе механизма внимания с использованием подхода машинного обучения seq2seq, где модель CNN на основе механизма внимания является кодировщиком, а двунаправленная модель LSTM декодировщиком соответственно. Предназначения используемых моделей можно представить следующим образом:

- CNN модель на основе механизма внимания извлекает глубинные характеристики временного ряда;
- LSTM модель добывает долгосрочные характеристики временного ряда;



- XGBoost модель используется для тонкой настройки, позволяя полностью получить информацию о фондовом рынке за несколько периодов.

Также важно отметить, что применение подхода машинного обучения seq2seq позволяет подавить влияние шума при помощи используемой архитектуры кодировщик-декодировщик. На основе глубокого обучения скрытая информация о состоянии отображается более эффективно, пока модель не будет удовлетворять предположениям о линейных свойствах цены акции. В целом, слои кодировщик-декодировщик отображают зависимость между нынешней и предыдущей последовательностью, а также между нынешней последовательностью и эмбедами.

Основным аргументом в пользу использования модели CNN на основе механизма внимания, в сочетании с моделью LSTM является тот факт, что CNN модель способна выявлять глобальные и локальные зависимости, которые не всегда улавливаются LSTM моделью. Причина описанного ранее явления кроется в том, что используемая модель CNN совмещается в себе, как и слои свертки, так и слои множественного внимания. В итоге совмещение LSTM модели с CNN моделью повышает, как и структурные преимущества, так и способности моделирования временных рядов.

После декодирования выходные данные попадают в XGBoost регрессор для более тщательного извлечения характеристик и тонкой настройки. Как модель для тонкой настройки XGBoost обладает рядом внушительных преимуществ, которыми являются расширяемость и гибкость. Данная модель интегрирует несколько различных модельных деревьев для получения более эффективной модели обучения, как итога XGBoost модель повышает конечные прогностические и обобщающие способности модели.

## 2.4 IDEF0-диаграмма

На рисунках 7 и 8 представлена диаграмма IDEF0 комбинированного метода прогнозирования временных рядов на фондовом рынке.

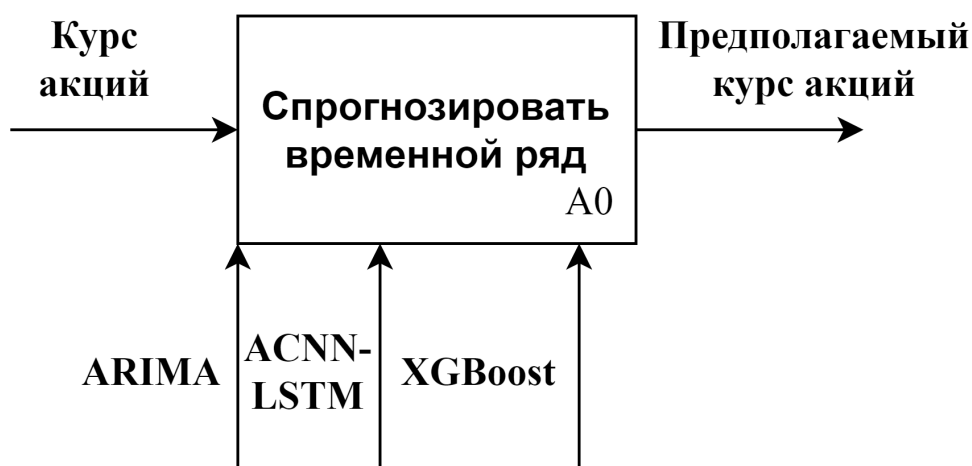


Рисунок 7 – Функциональная модель метода в виде IDEF0 — диаграммы уровня A0.

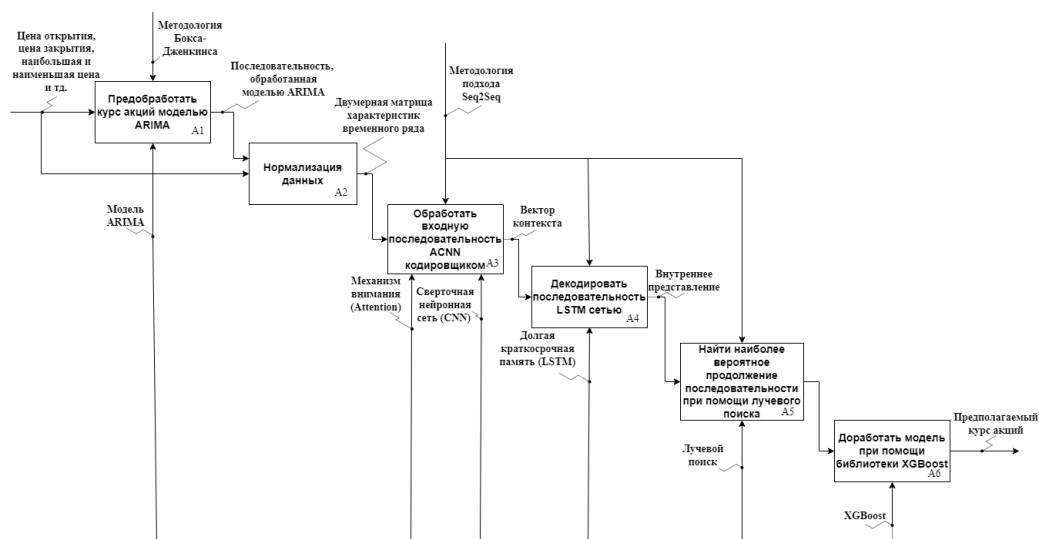


Рисунок 8 – Функциональная модель метода в виде IDEF0 — диаграммы уровня A1-A6.

## 2.5 Выводы из конструкторского раздела

В рамках данного раздела были углубленно изложены задействованные методы и алгоритмы, на их основе разработан и описан комбинированный ме-

тод, который затем был декомпозицирован. Основой разработанного комбинированного метода выступили такие классические модели, как ARIMA, ACNN, BiLSTM, исходя из предположения, что их интеграция в нелинейной зависимости способна улучшить эффективность предсказания временных рядов.

### 3 Технологический раздел

В данном разделе будет произведен выбор и обоснование языка программирования, среди разработки, а также задействованных библиотек. Помимо этого также будут описаны задействованные данные и разработана структура программного обеспечения.

#### 3.1 Выбор и обоснование языка программирования, среды разработки и задействованных библиотек

На следующем этапе исследования были реализованы прогнозные модели. Для этого использовался язык программирования Python. Интегрированной средой разработки был выбран Google Colaboratory – облачный сервис для проведения исследований в машинном и глубоком обучении, который представляет собой онлайн–блокнот Jupyter, не требующий дополнительных установок и позволяющий запускать код на графическом процессоре Nvidia Tesla V100, что значительно ускоряет процесс обучения моделей. В исследовании использовались следующие библиотеки:

- *Scikit-learn*. Данная библиотека содержит готовые реализации многих алгоритмов машинного обучения, а также метрики оценивания алгоритмов;
- *Matplotlib*. Использовалась для построения графиков;
- *Keras*. Нейросетевая библиотека, предназначенная для оперативной имплементации сетей глубокого обучения. Она содержит готовые слои и умеет объединять их между собой в сеть, обучать и реализовывать с ее помощью прогнозы;
- *TensorFlow*. Библиотека машинного и глубокого обучения позволяет оптимизировать вычисления, производимые во время обучения;
- *Pandas*. Библиотека предназначена для обработки и анализа данных;
- *XGBoost*. Библиотека предназначена для повышения градиента;

— *Numpy*. Библиотека для поддержки больших многомерных массивов данных.

### 3.2 Задействованные данные

В основе данной работы лежит эмпирическое исследование курса акций Bank of China на фондовом рынке. Данные о цене акций находятся в открытом доступе и отобраны в диапазоне с первого января 2007 года по 31 марта 2022 года. Один день торгов является по своей сущности единицей последовательности временного ряда. Сформированный на их основе тренировочный и тестовый набор отображены на рисунке под номером 9. Процентное соотношение тренировочного и тестового набора составляет 95% и 5% соответственно.



Рисунок 9 – Курс акций Bank of China, разбитый на тренировочный и тестовый набор.

Важно отметить, что данные о цене акций состоят из таких параметров, как цена открытия и закрытия, наибольшая и наименьшая цена за день и так далее. Затем эти данные проходят через модель ARIMA. Полученная в результате последовательность совместно с изначальными данными является характеристиками временного ряда, отображенная схематично на рисунке 10.

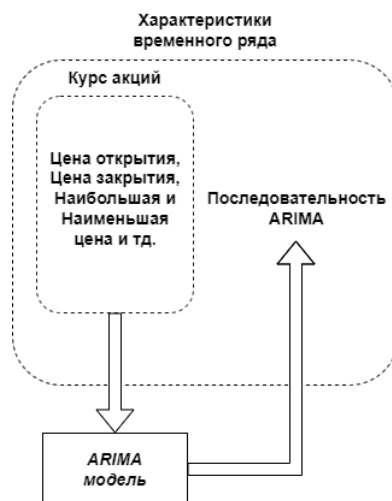


Рисунок 10 – Схема составных частей характеристик временного ряда.

Входными данными нейросетевого модуля является двумерная матрица размером количества характеристик временного ряда на временное окно. Шириной временного окна в данном случае является число 20.

### 3.3 Структура разработанного программного комплекса

Структура разработанного программного комплекса представлена на рисунке 11. Тренировка проходила в течение 50 эпох для обеспечения стабильного результата точности и потерь. В качестве функции оптимизации для модели была использована функция Adam (адаптивная оценка момента). Количество слоев LSTM 5 с размером в 64. Модель тренируется при помощи выбывания (dropout), уровень которого составляет 0.3.

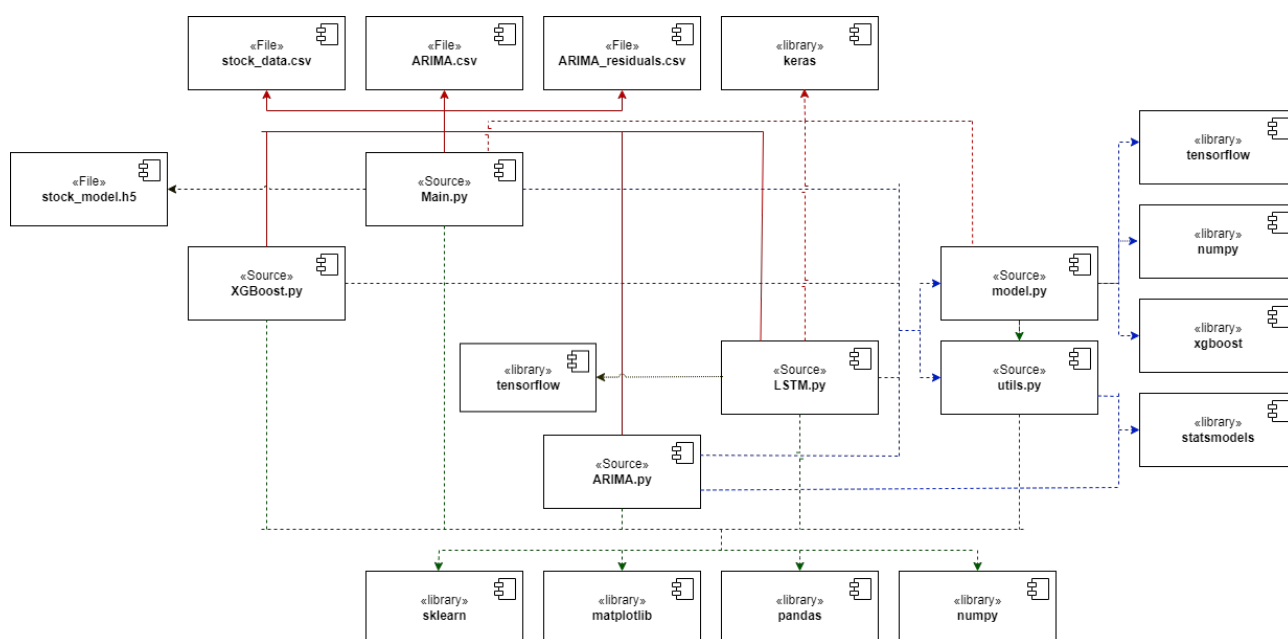


Рисунок 11 – Структура разработанного программного комплекса.

### **3.4 Выводы из технологического раздела**

В данном разделе был произведен выбор и обоснование языка программирования, среди разработки, а также задействованных библиотек. Помимо этого также были подробно описаны задействованные данные и разработана структура программного обеспечения.



## 4 Исследовательский раздел

В данном разделе будет исследована эффективность разработанного комбинированного метода прогнозирования временных рядов при различных сочетаниях используемых моделей. На основе полученных значений будет произведен сравнительный анализ комбинаций разработанных моделей. По результатам проведенного исследования будут сделаны выводы о точности предсказаний гибридной модели.

### 4.1 Эффективность предсказаний

#### ARIMA

Эффективность предсказаний курса акций при помощи классической модели ARIMA отображена на рисунке 12.



Рисунок 12 – Предсказанный и реальный курс акций с использованием ARIMA.

Остаток и плотность остатка предсказаний модели ARIMA показана на рисунке 13.

Разница между предсказанным и реальным значением курса акций на обучающем и проверочном наборе отображена на рисунке 14.

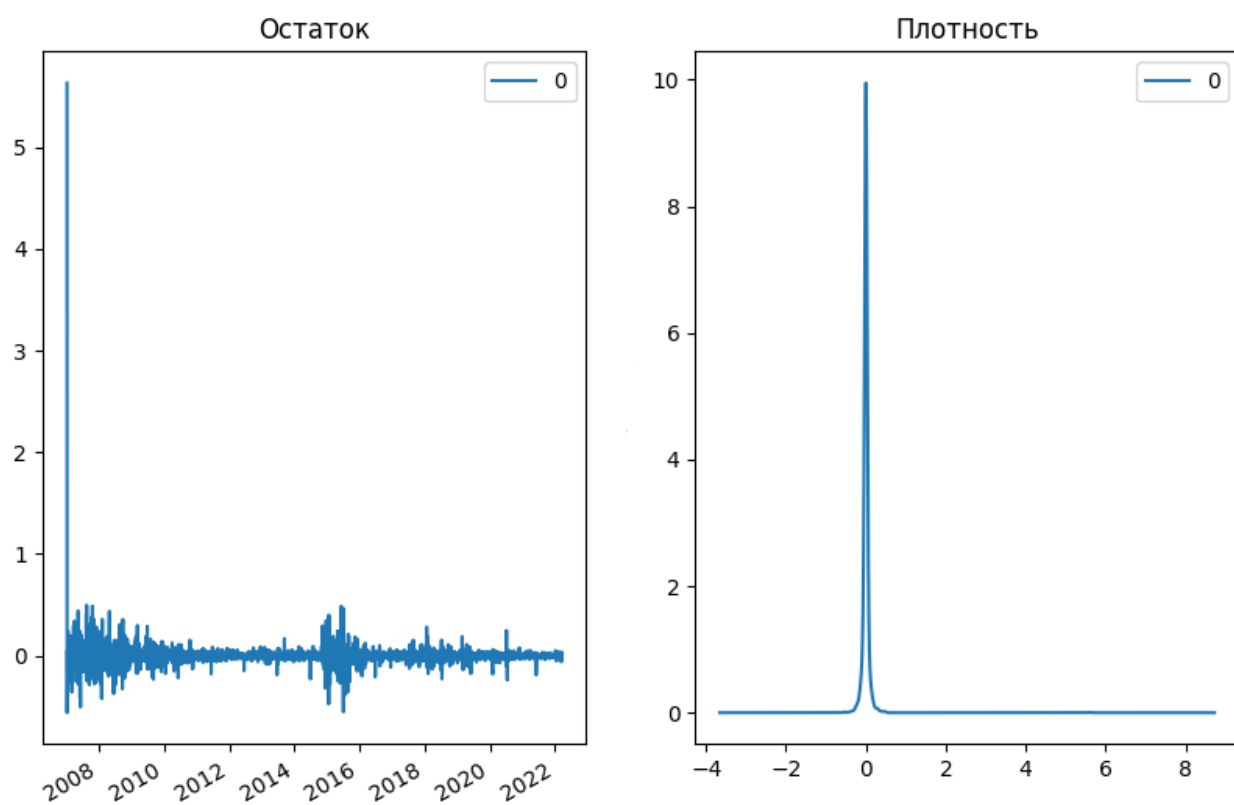


Рисунок 13 – Остаток и плотность остатка предсказаний модели ARIMA.

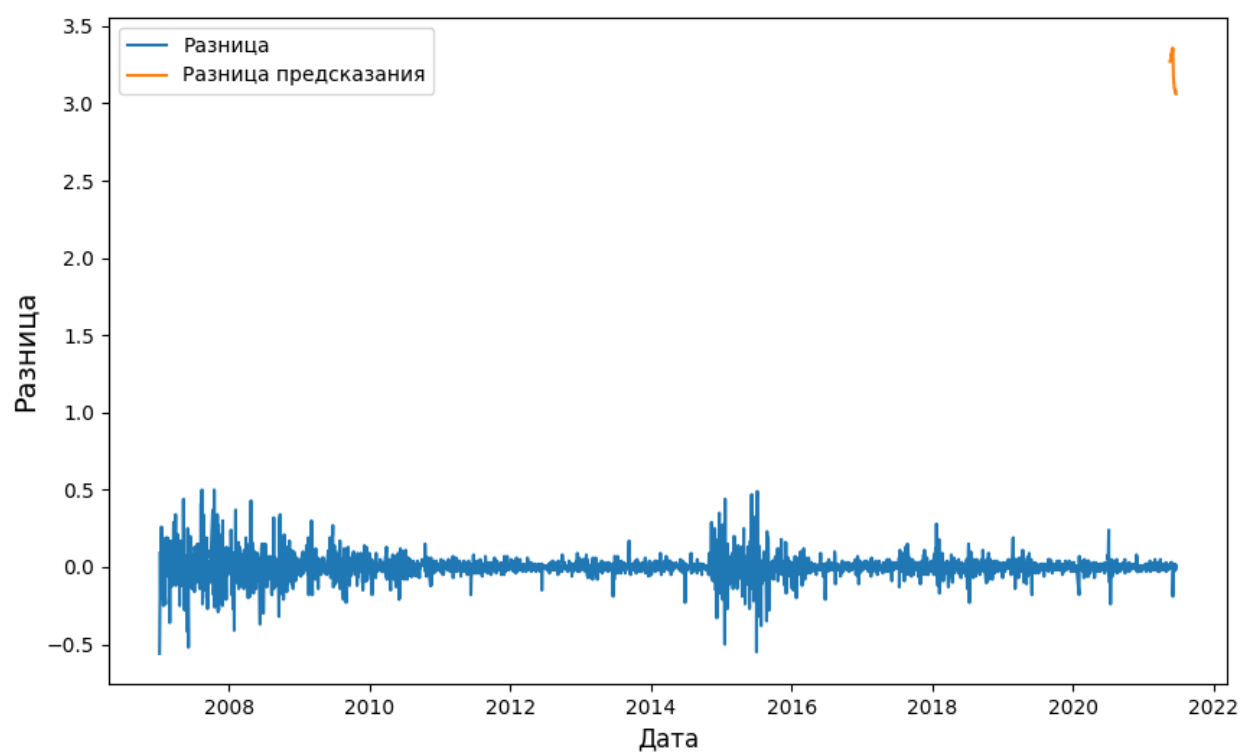


Рисунок 14 – Разница предсказаний модели ARIMA.

## ARIMA + XGBoost

Эффективность предсказаний курса акций при помощи классической модели ARIMA с дальнейшей обработкой XGBoost регрессором отображена на рисунке 15.



Рисунок 15 – Предсказанный и реальный курс акций моделью ARIMA+XGBoost.

Остаток предсказаний модели ARIMA + XGBoost показана на рисунке 16.

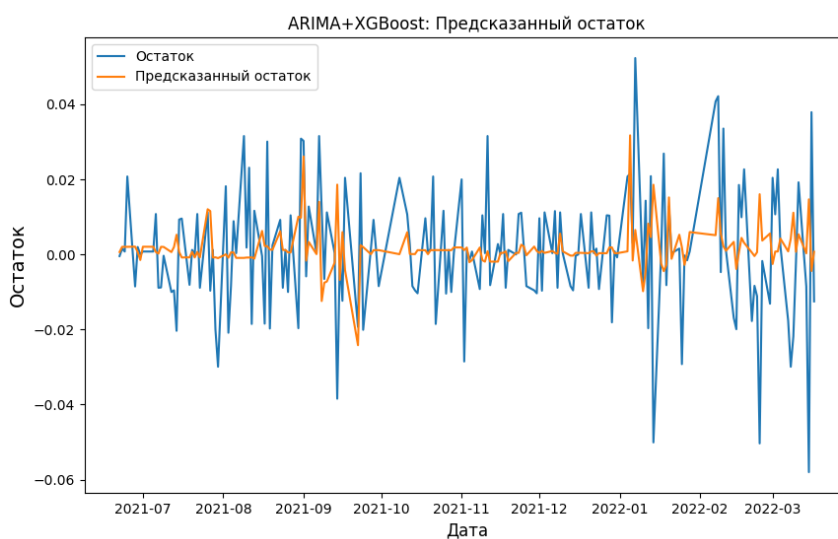


Рисунок 16 – Остаток предсказанный моделью ARIMA + XGBoost.

## ARIMA + LSTM

Кривые потерь исходной последовательности ARIMA + SingleLSTM показаны на рисунке 17.

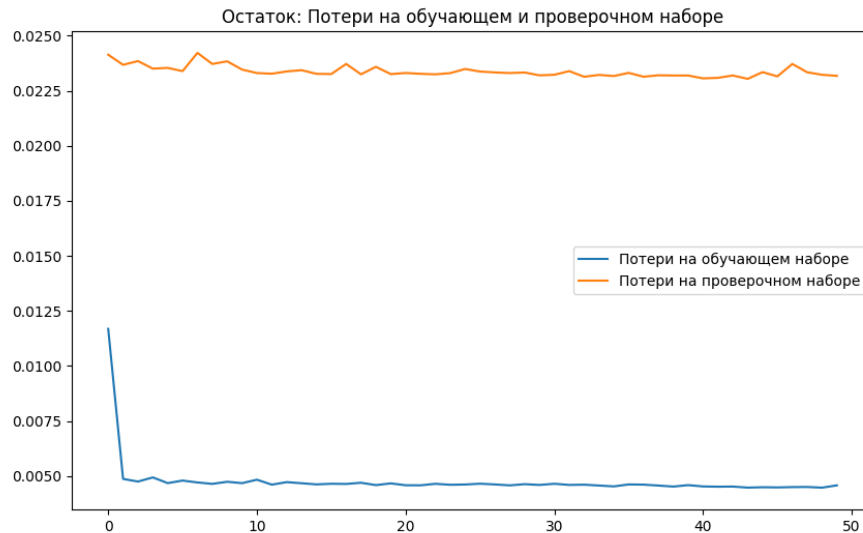


Рисунок 17 – Потери исходной последовательности ARIMA + SingleLSTM.

Кривые потерь остаточной последовательности ARIMA + SingleLSTM показаны на рисунке 18.

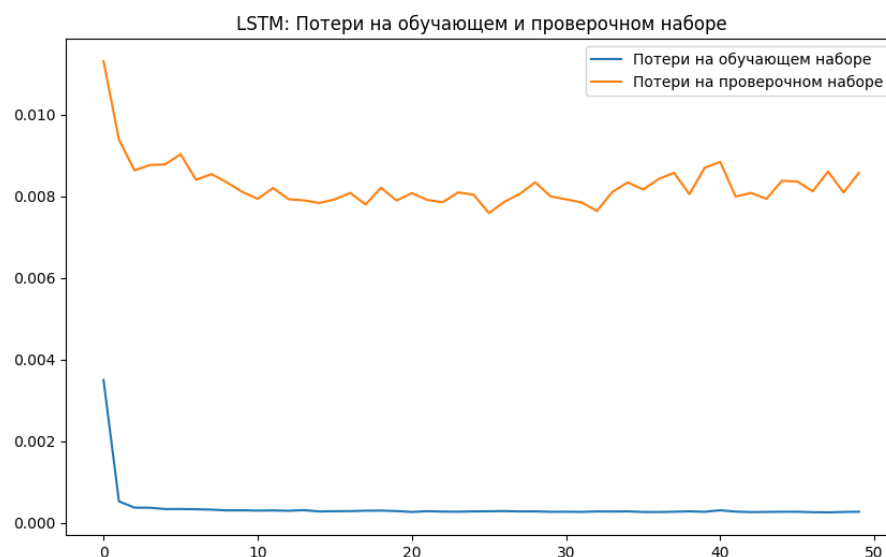


Рисунок 18 – Потери остаточной последовательности ARIMA + SingleLSTM.

Эффективность предсказаний курса акций модели ARIMA + SingleLSTM отображены на рисунке 19.

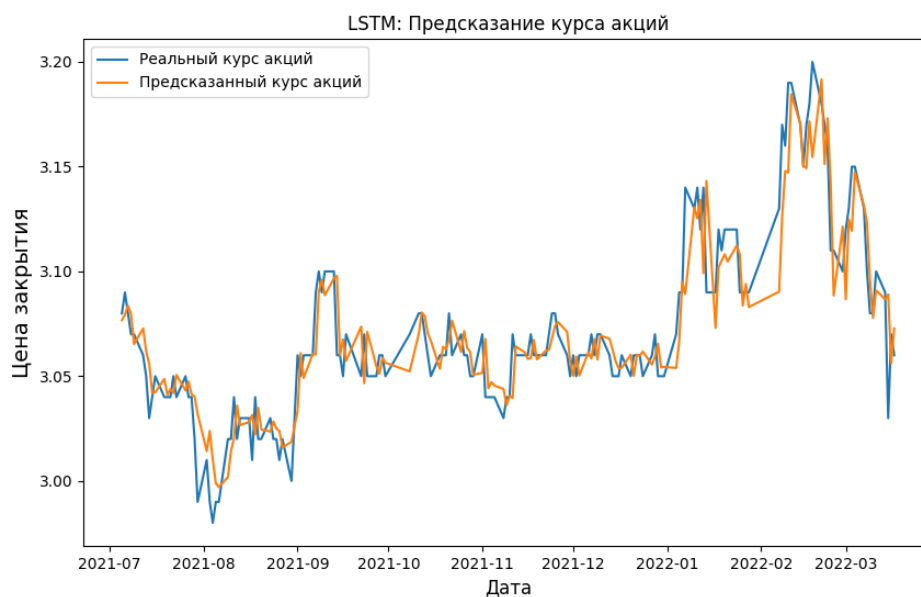


Рисунок 19 – Предсказанный курс акций моделью ARIMA + SingleLSTM.

Эффективность предсказания курса акций при помощи модели ARIMA + BiLSTM отображены на рисунке 20.



Рисунок 20 – Предсказанный курс акций моделью ARIMA + BiLSTM.

## ARIMA + ACNN + LSTM + XGBoost

Эффективность предсказания курса акций при помощи предложенной комбинированной модели ARIMA + ACNN + LSTM + XGBoost отображены на рисунке 21.

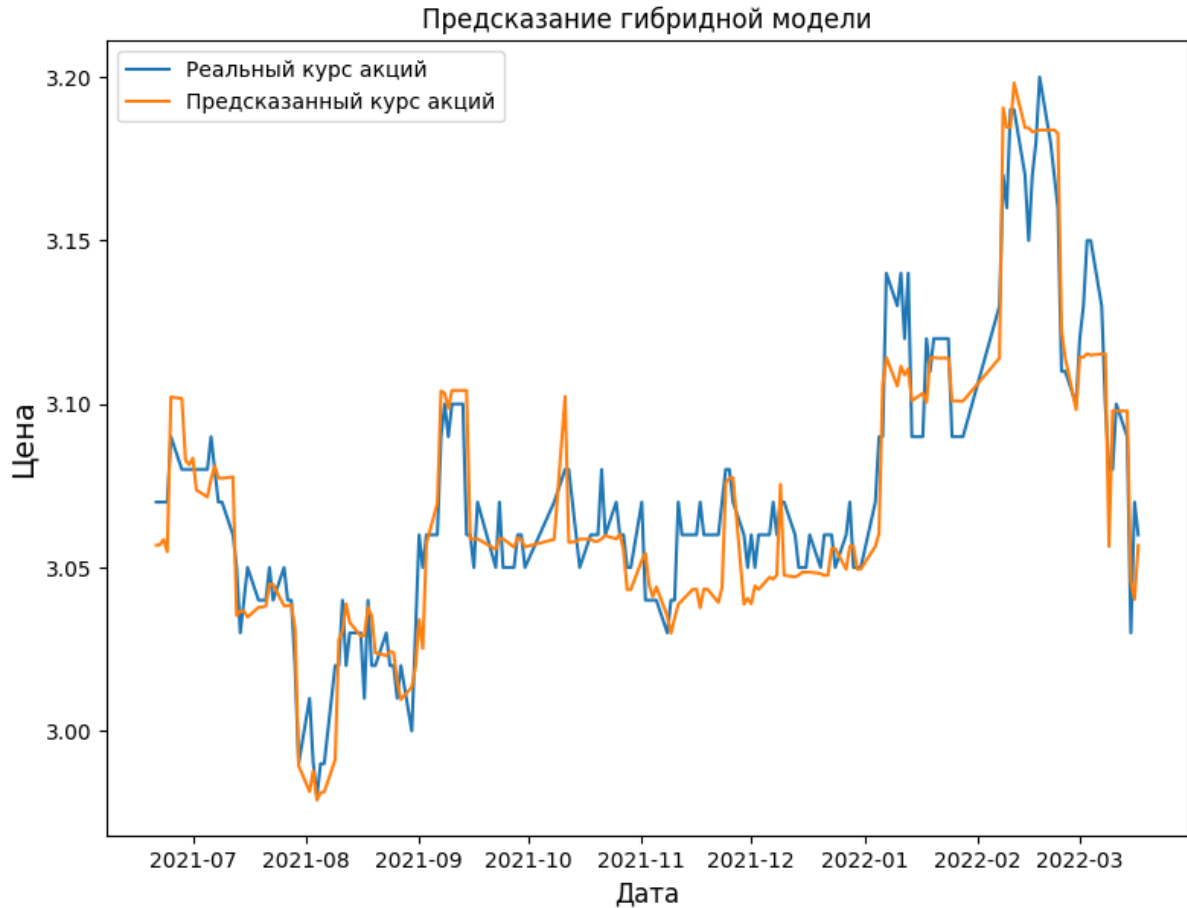


Рисунок 21 – Предсказанный курс акций гибридной моделью.

### 4.2 Сравнение комбинаций разработанных моделей

Метриками оценки комбинации разработанных моделей будут служить: средняя абсолютная ошибка (MAE, формула 13), корень среднеквадратичной ошибки (RMSE, формула 14), средняя абсолютная процентная ошибка (MAPE, формула 15), а также  $R^2$ , представленная в формуле 16.

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n \left| \hat{X}_t - X_t \right| \quad (13)$$

Здесь  $\bar{X}_t$  обозначает среднее значение  $X_t$ . Чем меньше ошибка и выше оценка  $R^2$ , тем лучше производительность соответственно.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (\hat{X}_t - X_t)^2} \quad (14)$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{\hat{X}_t - X_t}{X_t} \right| \quad (15)$$

$$R^2 = \frac{\sum_{t=1}^n \|\hat{X}_t - \bar{X}_t\|^2}{\sum_{t=1}^n \|X_t - \bar{X}_t\|^2} \quad (16)$$

Таблица 3 – Сравнение различных сочетаний моделей предобучения-дообучения

Предобучение	Дообучение	MSE	RMSE	MAE	R2
—	—	0.00057	0.02734	0.02368	0.74402
—	XGBoost	0.00031	0.01755	0.01223	0.82405
SL-LSTM	SL-LSTM	0.00045	0.02282	0.01960	0.79434
ML-LSTM	ML-LSTM	0.00031	0.01720	0.01265	0.82351
BiLSTM	BiLSTM	0.00027	0.01652	0.01201	0.84210
BiLSTM	XGBoost	0.00024	0.01605	0.01187	0.86301
CNN-BiLSTM	XGBoost	0.00022	0.01529	0.01145	0.87720
ACNN-BiLSTM	XGBoost	<b>0.00020</b>	<b>0.01424</b>	<b>0.01126</b>	<b>0.88342</b>

### 4.3 Выводы из исследовательского раздела

На основе проведенного исследования эффективности прогнозирования временных рядов при различных сочетаний моделей предобучения-дообучения можно утверждать, что разработанная концепция комбинированной модели превосходит, как и отдельно взятые авторегрессионные и нейросетевые аналоги, так и их простейшие комбинации.

## ЗАКЛЮЧЕНИЕ

Трудно недооценить влияние оказываемое фондовым рынком на темпы экономического и финансового развития любой отрасли. Тем не менее эффективности инфраструктуры рынка ценных бумаг противопоставляется сложная волатильность, как следствие возможность предсказания тенденций курса акций — это важнейшее стратегическое преимущество способное защитить, и потенциально преумножить вложения инвесторов.

К сожалению, классические модели временных рядов по типу ARIMA не в состоянии описать всю нелинейность прогнозирования фондового рынка, зато нейронные сети как раз обладают потенциалом на такую способность.

В этой работе основное внимание было отведено комбинированной модели на базе известных моделей ARIMA, CNN-LSTM и XGBoost. Их совместная интеграция в нелинейной зависимости позволила добиться заметного улучшения точности предсказания, нежели по-одиночке. Данная модель может фиксировать информацию о фондовом рынке за несколько периодов. Данные предварительно обрабатываются через модель ARIMA, затем архитектура глубокого обучения формируется в рамках предобучения-дообучения с применением фреймворков. Предобучение представляет собой CNN-LSTM модель на основе подхода seq2seq. Первоначально модель при помощи механизма внимания множественной свертки добывает глубинные характеристики временного ряда, затем, XGBoost регрессор применяется с целью дообучения комбинированной модели.

Результаты отчетливо демонстрируют, что концепция комбинированной модели превалирует над отдельно взятыми авторегрессионными и нейросетевыми аналогами, и как следствие является перспективной ветвью развития в вопросах прогнозирования временных рядов на фондовом рынке.

В результате выполнения выпускной квалификационной работы была достигнута поставленная цель — исследованы существующие решения, на их ос-



нове разработан комбинированный метод прогнозирования временных рядов на фондовом рынке. На пути к достижению цели были решены следующие задачи:

- Описаны основные понятия предметной области и обозначена проблема;
- Проведен анализ существующих методов и средств прогнозирования временных рядов;
- Отобраны наиболее удовлетворяющие параметрам эффективности методы.
- Выбранные методы разработаны;
- Разработано комбинирование реализованных методов;

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Число частных инвесторов на Московской бирже достигло 10 миллионов [Электронный ресурс]. – Режим доступа:  
*[https : //www.moex.com/n32746/?nt = 106](https://www.moex.com/n32746/?nt=106)*
2. Исследование Тинькофф Инвестиций: российский розничный инвестор — кто он такой? [Электронный ресурс]. – Режим доступа:  
*[https : //vc.ru/finance/107104 – issledovanie – tinkoff – investiciy – rossiyskiy – roznichnyy – investor – kto – on – takoy](https://vc.ru/finance/107104-issledovanie-tinkoff-investiciy-rossiyskiy-roznichnyy-investor-kto-on-takoy)* ( Дата обращения: 01.12.2021 )
3. Билл Вильямс. Торговый Хаос. Экспертные методики максимизации прибыли. «Аналитика», 2006 г., 330 с
4. В.Ю. Цибульников. Фундаментальный анализ рынка ценных бумаг. «Эль Контент», 2013 г., 99 с
5. 80 percent of the stock market is now on autopilot. [Электронный ресурс]. – Режим доступа:  
*[https : //www.cnn.com/2019/06/28/80percent – of – the – stock – market – is – now – on – autopilot.htm](https://www.cnn.com/2019/06/28/80percent-of-the-stock-market-is-now-on-autopilot.htm)* ( Дата обращения: 01.12.2021 )
6. Поведенческие модели участников биржи. [Электронный ресурс]. – Режим доступа:  
*[https : //www.hse.ru/data/2015/06/19/1084070439/dis.pdf](https://www.hse.ru/data/2015/06/19/1084070439/dis.pdf)* ( Дата обращения: 03.12.2021 )
7. Влияние корпоративных новостей на рыночную стоимость компаний. [Электронный ресурс]. – Режим доступа:  
*[http : //ecsocman.hse.ru/data/368/769/1223/Vypusk9\\_3.pdf](http://ecsocman.hse.ru/data/368/769/1223/Vypusk9_3.pdf)* ( Дата обращения: 06.12.2021 )

8. Влияние новостного фона на стоимость компаний. [Электронный ресурс]. – Режим доступа:  
*[https://www.jsdrm.ru/jour/article/view/805?locale=ru\\_RU](https://www.jsdrm.ru/jour/article/view/805?locale=ru_RU)* ( Дата обращения: 06.12.2021 )
9. Is “Scalping” Irrational? [Электронный ресурс]. – Режим доступа:  
*<https://financial-hacker.com/is-scalping-irrational/>* ( Дата обращения: 02.12.2021 )
10. Анализ и прогнозирование финансового рынка на основе модели детерминированного хаоса. [Электронный ресурс]. – Режим доступа:  
*[http://earchive.tpu.ru/bitstream/11683/6337/1/thesis\\_tpu-2007-23.pdf](http://earchive.tpu.ru/bitstream/11683/6337/1/thesis_tpu-2007-23.pdf)*  
( Дата обращения: 05.12.2021 )
11. A Time Series Analysis-Based Stock Price Prediction Framework Using Machine Learning and Deep Learning Models. [Электронный ресурс]. – Режим доступа:  
*[https://www.researchgate.net/publication/340720727\\_A\\_Time\\_Series\\_Analysis-Based\\_Stock\\_Price\\_Prediction\\_Framework\\_Using\\_Machine\\_Learning\\_and\\_Deep\\_Learning\\_Model](https://www.researchgate.net/publication/340720727_A_Time_Series_Analysis-Based_Stock_Price_Prediction_Framework_Using_Machine_Learning_and_Deep_Learning_Model)* ( Дата обращения: 05.12.2021 )
12. Adaptive filter design for stock market prediction using a correlation-based criterion. [Электронный ресурс]. – Режим доступа:  
*<https://arxiv.org/ftp/arxiv/papers/1501/1501.07504.pdf>*  
*[5l7tp33v20/direct/176401688.pdf](https://arxiv.org/ftp/arxiv/papers/1501/1501.07504.pdf)* ( Дата обращения: 06.12.2021 )
13. Prediction of Stock Market Index based on Neural Networks, Genetic Algorithms, Data Mining Using SVD. [Электронный ресурс]. – Режим доступа:  
*[https://www.researchgate.net/publication/279983540\\_Prediction\\_of\\_Stock\\_Market\\_Index\\_based\\_on\\_Neural\\_Networks\\_Genetic\\_Algorithms\\_and\\_Data\\_Mining\\_Using\\_SVD](https://www.researchgate.net/publication/279983540_Prediction_of_Stock_Market_Index_based_on_Neural_Networks_Genetic_Algorithms_and_Data_Mining_Using_SVD)* ( Дата обращения: 05.12.2021 )

14. Prediction of Stock Market Index Using Genetic Algorithm. [Электронный ресурс]. – Режим доступа:  
*https : //core.ac.uk/download/pdf/234644513.pdf* ( Дата обращения: 05.12.2021 )
15. Machine Learning Techniques for Stock Prediction. [Электронный ресурс]. – Режим доступа:  
*https : //docplayer.net/4473092 – Machine – learning – techniques – for – stock – prediction – vatsal – h – shah.html* ( Дата обращения: 02.12.2021 )
16. A Framework for Predictive Analysis of Stock Market Indices – A Study of the Indian Auto Sector. [Электронный ресурс]. – Режим доступа:  
*https : //arxiv.org/ftp/arxiv/papers/1604/1604.04044.pdf* ( Дата обращения: 03.12.2021 )
17. Combining Time Series and Sentiment Analysis for Stock Market Forecasting. [Электронный ресурс]. – Режим доступа:  
*https : //avestia.com/ICSTA2021\_proceedings/files/paper/ICSTA\_32.pdf* ( Дата обращения: 04.12.2021 )
18. Short-term stock market price trend prediction using a comprehensive deep learning system. [Электронный ресурс]. – Режим доступа:  
*https : //journalofbigdata.springeropen.com/track/pdf/10.1186/s40537 – 020 – 00333 – 6.pdf* ( Дата обращения: 04.12.2021 )
19. Stock Price Forecast Based on CNN-BiLSTM-ECA Model. [Электронный ресурс]. - Режим доступа  
*https : //www.hindawi.com/journals/sp/2021/2446543/* ( Дата обращения: 05.12.2021 )
20. Прогнозирование динамики курсовых стоимостей акций фондового рынка с применением резонансных систем искусственного интеллекта. [Элек-

- тронный ресурс]. – Режим доступа: [https : //publications.hse.ru/mirror/pubs/share/folder/5l7tp33v20/direct/176401688.pdf](https://publications.hse.ru/mirror/pubs/share/folder/5l7tp33v20/direct/176401688.pdf) ( Дата обращения: 06.12.2021 )
21. Финансовые временные ряды: прогнозирование и распознавание нарушений динамики. [Электронный ресурс]. – Режим доступа: [http : //math.nsc.ru/conference/zont09/reports/09Demin – Vityaev.pdf](http://math.nsc.ru/conference/zont09/reports/09Demin-Vityaev.pdf) ( Дата обращения: 08.12.2021 )
22. Эффективность методов фундаментального и технического анализа для совершения инвестиционных операций [Электронный ресурс]. – Режим доступа: [https : //publications.hse.ru/pubs/share/folder/c8e5n64vy8/77877838.Pdf](https://publications.hse.ru/pubs/share/folder/c8e5n64vy8/77877838.Pdf) ( Дата обращения: 08.12.2021 )
23. Chaos-based support vector regressions for exchange rate forecasting [Электронный ресурс]. – Режим доступа: [https : //www.researchgate.net/publication/220216857\\_Chaos-based\\_support\\_vector\\_regressions\\_for\\_exchange\\_rate\\_forecasting](https://www.researchgate.net/publication/220216857_Chaos-based_support_vector_regressions_for_exchange_rate_forecasting)2022030108413520220301084135 ( Дата обращения: 08.12.2021 )
24. Чучуева И.А. МОДЕЛЬ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ ПО ВЫБОРКЕ МАКСИМАЛЬНОГО ПОДОБИЯ, диссертация... канд. тех. наук / Московский государственный технический университет им. Н.Э. Баумана. Москва, 2012.
25. Jingfei Yang M. Sc. Power System Short-term Load Forecasting: Thesis for Ph.d degree. Germany, Darmstadt, Elektrotechnik und Informationstechnik der Technischen Universität, 2006. 139 p.
26. Collantes-Duarte J., Rivas-Echeverriat F. Time Series Forecasting using ARIMA, Neural Networks and Neo Fuzzy Neurons // WSEAS International

- Conference on Neural Networks and Applications, Switzerland, 2002 [Электронный ресурс]. – Режим доступа: [www.wseas.us/e-library/conferences/switzerland2002/papers/464.pdf](http://www.wseas.us/e-library/conferences/switzerland2002/papers/464.pdf) (Дата обращения: 20.05.2022).
27. Alfares H.K., Nazeeruddin M. Electric load forecasting: literature survey and classification of methods // International Journal of Systems Science. 2002, Vol 33. P. 23 – 34.
28. A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, “Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting,” [Электронный ресурс]. – Режим доступа: <https://ieeexplore.ieee.org/document/9057460> (Дата обращения: 20.05.2022).
29. A CNN-LSTM-Based Model to Forecast Stock Prices [Электронный ресурс]. – Режим доступа: <https://www.hindawi.com/journals/complexity/2020/6622927/> (Дата обращения: 21.05.2022).
30. A. Graves, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

## ПРИЛОЖЕНИЕ А

### Листинг 1: Разработанная комбинированная модель

```
1 from keras.layers import Input, Dense, LSTM, Conv1D, Dropout, Bidirectional,
    Multiply
2 from keras.models import Model
3 # from attention_utils import get_activations
4 from keras.layers import merge
5 from keras.layers.core import *
6 from keras.layers.recurrent import LSTM
7 from keras.models import *
8 from tensorflow.keras import backend as K
9 from utils import *
10 import numpy as np
11 import xgboost as xgb
12
13 def attention_3d_block_merge(inputs, single_attention_vector = False):
14     # inputs.shape = (batch_size, time_steps, input_dim)
15     input_dim = int(inputs.shape[2])
16     a = inputs
17     # a = Permute((2, 1))(inputs)
18     # a = Reshape((input_dim, TIME_STEPS))(a)
19     a = Dense(input_dim, activation='softmax')(a)
20     if single_attention_vector:
21         a = Lambda(lambda x: K.mean(x, axis=1), name='dim_reduction')(a)
22         a = RepeatVector(input_dim)(a)
23     a_probs = Permute((1, 2), name='attention_vec')(a)
24
25     output_attention_mul = merge([inputs, a_probs], name='attention_mul',
        mode='mul')
26     return output_attention_mul
27
28 def attention_3d_block(inputs, single_attention_vector=False):
29     # inputs.shape = (batch_size, time_steps, input_dim)
30     time_steps = K.int_shape(inputs)[1]
31     input_dim = K.int_shape(inputs)[2]
32     a = Permute((2, 1))(inputs)
33     a = Dense(time_steps, activation='softmax')(a)
34     if single_attention_vector:
```

```

35         a = Lambda(lambda x: K.mean(x, axis=1))(a)
36         a = RepeatVector(input_dim)(a)
37
38         a_probs = Permute((2, 1))(a)
39         output_attention_mul = Multiply()([inputs, a_probs])
40         return output_attention_mul
41
42 def attention_model(INPUT_DIMS = 13, TIME_STEPS = 20, lstm_units = 64):
43     inputs = Input(shape=(TIME_STEPS, INPUT_DIMS))
44
45     x = Conv1D(filters=64, kernel_size=1, activation='relu')(inputs) #
padding = 'same'
46     x = Dropout(0.3)(x)
47
48     # lstm_out = Bidirectional(LSTM(lstm_units, activation='relu'), name='
bilstm')(x)
49     lstm_out = Bidirectional(LSTM(lstm_units, return_sequences=True))(x)
50     lstm_out = Dropout(0.3)(lstm_out)
51     attention_mul = attention_3d_block(lstm_out)
52     attention_mul = Flatten()(attention_mul)
53
54     output = Dense(1, activation='sigmoid')(attention_mul)
55     model = Model(inputs=[inputs], outputs=output)
56     return model
57
58 def PredictWithData(data, data_yuan, name, modelname, INPUT_DIMS = 13, TIME_STEPS
= 20):
59     print(data.columns)
60     yindex = data.columns.get_loc(name)
61     data = np.array(data, dtype='float64')
62     data, normalize = NormalizeMult(data)
63     data_y = data[:, yindex]
64     data_y = data_y.reshape(data_y.shape[0], 1)
65
66     testX, _ = create_dataset(data)
67     _, testY = create_dataset(data_y)
68     print("testX Y shape is:", testX.shape, testY.shape)
69     if len(testY.shape) == 1:
70         testY = testY.reshape(-1, 1)
71

```



```

72     model = attention_model(INPUT_DIMS)
73     model.load_weights(modelname)
74     model.summary()
75     y_hat = model.predict(testX)
76     testY, y_hat = xgb_scheduler(data_yuan, y_hat)
77     return y_hat, testY
78
79 def lstm(model_type,X_train,yuan_X_train):
80     if model_type == 1:
81         # single-layer LSTM
82         model = Sequential()
83         model.add(LSTM(units=50, activation='relu',
84             input_shape=(X_train.shape[1], 1)))
85         model.add(Dense(units=1))
86         yuan_model = Sequential()
87         yuan_model.add(LSTM(units=50, activation='relu',
88             input_shape=(yuan_X_train.shape[1], 5)))
89         yuan_model.add(Dense(units=5))
90     if model_type == 2:
91         # multi-layer LSTM
92         model = Sequential()
93         model.add(LSTM(units=50, activation='relu', return_sequences=True,
94             input_shape=(X_train.shape[1], 1)))
95         model.add(LSTM(units=50, activation='relu'))
96         model.add(Dense(1))
97
98         yuan_model = Sequential()
99         yuan_model.add(LSTM(units=50, activation='relu', return_sequences=
100 True,
101             input_shape=(yuan_X_train.shape[1], 5)))
102         yuan_model.add(LSTM(units=50, activation='relu'))
103         yuan_model.add(Dense(5))
104     if model_type == 3:
105         # BiLSTM
106         model = Sequential()
107         model.add(Bidirectional(LSTM(50, activation='relu'),
108             input_shape=(X_train.shape[1], 1)))
109         model.add(Dense(1))
110
111         yuan_model = Sequential()

```

```

111         yuan_model.add(Bidirectional(LSTM(50, activation='relu'),
112                                     input_shape=(yuan_X_train.shape[1], 5)))
113         yuan_model.add(Dense(5))
114
115     return model, yuan_model
116
117 def xgb_scheduler(data, y_hat):
118     close = data.pop('close')
119     data.insert(5, 'close', close)
120     train, test = prepare_data(data, n_test=len(y_hat), n_in=6, n_out=1)
121     testY, y_hat2 = walk_forward_validation(train, test)
122     return testY, y_hat2
123
124 def xgboost_forecast(train, testX):
125     train = np.asarray(train)
126     # print('train', train)
127     trainX, trainy = train[:, :-1], train[:, -1]
128     # print('trainX', trainX, 'trainy', trainy)
129     model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=20)
130     model.fit(trainX, trainy)
131     yhat = model.predict(np.asarray([testX]))
132     return yhat[0]
133
134 def walk_forward_validation(train, test):
135     predictions = list()
136     train = train.values
137     history = [x for x in train]
138     # print('history', history)
139     for i in range(len(test)):
140         testX, testy = test.iloc[i, :-1], test.iloc[i, -1]
141         # print('i', i, testX, testy)
142         yhat = xgboost_forecast(history, testX)
143         predictions.append(yhat)
144         history.append(test.iloc[i, :])
145         print(i+1, '>expected=%.6f, predicted=%.6f' % (testy, yhat))
146     return test.iloc[:, -1], predictions

```

## Листинг 2: ARIMA модуль

```

1 import pandas as pd
2 import matplotlib.pyplot as plt

```

```

3 from datetime import datetime
4 import numpy as np
5 from pandas.plotting import autocorrelation_plot
6 from statsmodels.tsa.arima_model import ARIMA
7 from statsmodels.stats.diagnostic import acorr_ljungbox
8 from sklearn import metrics
9 from utils import *
10 plt.rcParams['font.sans-serif'] = ['SimHei']
11 plt.rcParams['axes.unicode_minus'] = False
12
13 data = pd.read_csv('./601988.SH.csv')
14 test_set2 = data.loc[3501:, :]
15 data.index = pd.to_datetime(data['trade_date'], format='%Y%m%d')
16 data = data.drop(['ts_code', 'trade_date'], axis=1)
17 data = pd.DataFrame(data, dtype=np.float64)
18
19 training_set = data.loc['2007-01-04':'2021-06-21', :] # 3501
20 test_set = data.loc['2021-06-22':, :] # 180
21
22 plt.figure(figsize=(10, 6))
23 plt.plot(training_set['close'], label='Тренировочный набор')
24 plt.plot(test_set['close'], label='Тестовый')
25 plt.title('Цена закрытия по датам')
26 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
27 plt.ylabel('Цена закрытия', fontsize=14, horizontalalignment='center')
28 plt.legend()
29 plt.show()
30
31 temp = np.array(training_set['close'])
32
33 training_set['diff_1'] = training_set['close'].diff(1)
34 plt.figure(figsize=(10, 6))
35 training_set['diff_1'].plot()
36 plt.title('Разница первого порядка')
37 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
38 plt.ylabel('Разница_1', fontsize=14, horizontalalignment='center')
39 plt.show()
40
41 training_set['diff_2'] = training_set['diff_1'].diff(1)
42 plt.figure(figsize=(10, 6))

```

```

43 training_set['diff_2'].plot()
44 plt.title('Разница второго порядка')
45 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
46 plt.ylabel('Разница_2', fontsize=14, horizontalalignment='center')
47 plt.show()
48
49 temp1 = np.diff(training_set['close'], n=1)
50
51 # Тест на белый шум
52 training_data1 = training_set['close'].diff(1)
53 # training_data1_nona = training_data1.dropna()
54 temp2 = np.diff(training_set['close'], n=1)
55 # print(acorr_ljungbox(training_data1_nona, lags=2, boxpierce=True, return_df
    =True))
56 print(acorr_ljungbox(temp2, lags=2, boxpierce=True))
57 # p-value=1.53291527e-08, временной ряд без белого шума
58
59 acf_pacf_plot(training_set['close'], acf_lags=160)
60
61 price = list(temp2)
62 data2 = {
63     'trade_date': training_set['diff_1'].index[1:],
64     'close': price
65 }
66
67 df = pd.DataFrame(data2)
68 df['trade_date'] = pd.to_datetime(df['trade_date'], format='%Y%m%d')
69
70 training_data_diff = df.set_index(['trade_date'], drop=True)
71 print('&', training_data_diff)
72
73 acf_pacf_plot(training_data_diff)
74
75 # order=(p,d,q) k
76 model = sm.tsa.ARIMA(endog=training_set['close'], order=(2, 1, 0)).fit()
77 #print(model.summary())
78
79 history = [x for x in training_set['close']]
80 # print('history', type(history), history)
81 predictions = list()

```

```

82 # print('test_set.shape', test_set.shape[0])
83 for t in range(test_set.shape[0]):
84     model1 = sm.tsa.ARIMA(history, order=(2, 1, 0))
85     model_fit = model1.fit()
86     yhat = model_fit.forecast()
87     yhat = np.float(yhat[0])
88     predictions.append(yhat)
89     obs = test_set2.iloc[t, 5]
90     # obs = np.float(obs)
91     # print('obs', type(obs))
92     history.append(obs)
93     # print(test_set.index[t])
94     # print(t+1, 'predicted=%f, expected=%f' % (yhat, obs))
95 #print('predictions', predictions)
96
97 predictions1 = {
98     'trade_date': test_set.index[:],
99     'close': predictions
100 }
101 predictions1 = pd.DataFrame(predictions1)
102 predictions1 = predictions1.set_index(['trade_date'], drop=True)
103 predictions1.to_csv('./ARIMA.csv')
104 plt.figure(figsize=(10, 6))
105 plt.plot(test_set['close'], label='Реальный курс акции')
106 plt.plot(predictions1, label='Предсказанный курс акции')
107 plt.title('ARIMA: Предсказание курса акций')
108 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
109 plt.ylabel('Цена закрытия', fontsize=14, horizontalalignment='center')
110 plt.legend()
111 plt.show()
112
113 model2 = sm.tsa.ARIMA(endog=data['close'], order=(2, 1, 0)).fit()
114 residuals = pd.DataFrame(model2.resid)
115 fig, ax = plt.subplots(1, 2)
116 residuals.plot(title="Остаток", ax=ax[0])
117 residuals.plot(kind='kde', title='Плотность', ax=ax[1])
118 plt.show()
119 residuals.to_csv('./ARIMA_residuals1.csv')
120 evaluation_metric(test_set['close'], predictions)
121 adf_test(temp)

```

```

122 adf_test(temp1)
123
124 predictions_ARIMA_diff = pd.Series(model.fittedvalues, copy=True)
125 predictions_ARIMA_diff = predictions_ARIMA_diff[3479:]
126 print(' ', predictions_ARIMA_diff)
127 plt.figure(figsize=(10, 6))
128 plt.plot(training_data_diff, label="Разница")
129 plt.plot(predictions_ARIMA_diff, label="Разница предсказания")
130 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
131 plt.ylabel('Разница', fontsize=14, horizontalalignment='center')
132 plt.title('DiffFit')
133 plt.legend()
134 plt.show()

```

### Листинг 3: LSTM модуль

```

1 import pandas as pd
2 import tensorflow as tf
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from sklearn import metrics
6 from sklearn.preprocessing import MinMaxScaler
7 from keras.models import Sequential
8 from keras.layers import Dense, LSTM, Bidirectional
9 from keras.optimizers import Adam
10 # from tensorflow.keras.optimizers import Adam
11 from numpy.random import seed
12 from utils import *
13 from model import lstm
14
15 # GPU
16 gpus = tf.config.experimental.list_physical_devices("GPU")
17 if gpus:
18     tf.config.experimental.set_memory_growth(gpus[0], True)
19     tf.config.set_visible_devices([gpus[0]], "GPU")
20
21 seed(1)
22 tf.random.set_seed(1)
23
24 n_timestamp = 10
25 n_epochs = 50

```

```

26 # =====
27 #      типы моделей
28 #      1. single-layer LSTM
29 #      2. multi-layer LSTM
30 #      3. bidirectional LSTM
31 # =====
32 model_type = 3
33
34 yuan_data = pd.read_csv('./601988.SH.csv')
35 yuan_data.index = pd.to_datetime(yuan_data['trade_date'], format='%Y%m%d')
36 yuan_data = yuan_data.loc[:, ['open', 'high', 'low', 'close', 'amount']]
37
38 data = pd.read_csv('./ARIMA_residuals1.csv')
39 data.index = pd.to_datetime(data['trade_date'])
40 data = data.drop('trade_date', axis=1)
41 # data = pd.merge(data, yuan_data, on='trade_date')
42
43 Lt = pd.read_csv('./ARIMA.csv')
44 idx = 3500
45 training_set = data.iloc[1:idx, :]
46 test_set = data.iloc[idx:, :]
47 yuan_training_set = yuan_data.iloc[1:idx, :]
48 yuan_test_set = yuan_data.iloc[idx:, :]
49
50 sc = MinMaxScaler(feature_range=(0, 1))
51 yuan_sc = MinMaxScaler(feature_range=(0, 1))
52 training_set_scaled = sc.fit_transform(training_set)
53 testing_set_scaled = sc.fit_transform(test_set)
54 yuan_training_set_scaled = yuan_sc.fit_transform(yuan_training_set)
55 yuan_testing_set_scaled = yuan_sc.fit_transform(yuan_test_set)
56
57 X_train, y_train = data_split(training_set_scaled, n_timestamp)
58 yuan_X_train, yuan_y_train = data_split(yuan_training_set_scaled, n_timestamp
    )
59 X_train = X_train.reshape(X_train.shape[0], X_train.shape[1], 1)
60 yuan_X_train = yuan_X_train.reshape(yuan_X_train.shape[0], yuan_X_train.shape
    [1], 5)
61
62 X_test, y_test = data_split(testing_set_scaled, n_timestamp)
63

```

```

64 X_test = X_test.reshape(X_test.shape[0], X_test.shape[1], 1)
65 yuan_X_test, yuan_y_test = data_split(yuan_testing_set_scaled, n_timestamp)
66 yuna_X_test = yuan_X_test.reshape(yuan_X_test.shape[0], yuan_X_test.shape[1],
    5)
67
68 model, yuan_model = lstm(model_type,X_train,yuan_X_train)
69 print(model.summary())
70 adam = Adam(learning_rate=0.01)
71 model.compile(optimizer=adam,
72               loss='mse')
73 yuan_model.compile(optimizer=adam,
74                   loss='mse')
75
76 history = model.fit(X_train, y_train,
77                     batch_size=32,
78                     epochs=n_epochs,
79                     validation_data=(X_test, y_test),
80                     validation_freq=1)
81 yuan_history = yuan_model.fit(yuan_X_train, yuan_y_train,
82                               batch_size=32,
83                               epochs=n_epochs,
84                               validation_data=(yuan_X_test, yuan_y_test),
85                               validation_freq=1)
86
87
88 plt.figure(figsize=(10, 6))
89 plt.plot(history.history['loss'], label='Потери на обучающем наборе')
90 plt.plot(history.history['val_loss'], label='Потери на проверочном наборе')
91 plt.title('Остаток: Потери на обучающем и проверочном наборе')
92 plt.legend()
93 plt.show()
94
95 plt.figure(figsize=(10, 6))
96 plt.plot(yuan_history.history['loss'], label='Потери на обучающем наборе')
97 plt.plot(yuan_history.history['val_loss'], label='Потери на проверочном
    наборе')
98 plt.title('LSTM: Потери на обучающем и проверочном наборе')
99 plt.legend()
100 plt.show()
101

```



```

102 yuan_predicted_stock_price = yuan_model.predict(yuan_X_test)
103 yuan_predicted_stock_price = yuan_sc.inverse_transform(
    yuan_predicted_stock_price)
104 yuan_predicted_stock_price_list = np.array(yuan_predicted_stock_price[:, 3]).
    flatten().tolist()
105 yuan_predicted_stock_price1 = {
106     'trade_date': yuan_data.index[idx+10:],
107     'close': yuan_predicted_stock_price_list
108 }
109 yuan_predicted_stock_price1 = pd.DataFrame(yuan_predicted_stock_price1)
110 yuan_predicted_stock_price1 = yuan_predicted_stock_price1.set_index(['
    trade_date'], drop=True)
111 yuan_real_stock_price = yuan_sc.inverse_transform(yuan_y_test)
112 yuan_real_stock_price_list = np.array(yuan_real_stock_price[:, 3]).flatten().
    tolist()
113 yuan_real_stock_price1 = {
114     'trade_date': yuan_data.index[idx+10:],
115     'close': yuan_real_stock_price_list
116 }
117 yuan_real_stock_price1 = pd.DataFrame(yuan_real_stock_price1)
118 yuan_real_stock_price1 = yuan_real_stock_price1.set_index(['trade_date'],
    drop=True)
119
120 predicted_stock_price = model.predict(X_test)
121 predicted_stock_price = sc.inverse_transform(predicted_stock_price)
122 predicted_stock_price_list = np.array(predicted_stock_price[:, 0]).flatten().
    tolist()
123
124 predicted_stock_price1 = {
125     'trade_date': data.index[idx+10:],
126     'close': predicted_stock_price_list
127 }
128 predicted_stock_price1 = pd.DataFrame(predicted_stock_price1)
129
130 predicted_stock_price1 = predicted_stock_price1.set_index(['trade_date'],
    drop=True)
131
132 real_stock_price = sc.inverse_transform(y_test)
133 finalpredicted_stock_price = pd.concat([Lt, predicted_stock_price1]).groupby(
    'trade_date')['close'].sum().reset_index()

```

```

134 finalpredicted_stock_price.index = pd.to_datetime(finalpredicted_stock_price[
    'trade_date'])
135 finalpredicted_stock_price = finalpredicted_stock_price.drop(['trade_date'],
    axis=1)
136
137 plt.figure(figsize=(10, 6))
138 # print('yuan_real', yuan_real_stock_price1)
139 plt.plot(yuan_data.loc['2021-06-22':, 'close'], label='Реальный курс акций')
140 plt.plot(finalpredicted_stock_price['close'], label='Предсказанный курс
    акций')
141 plt.title('BiLSTM: Предсказание курса акций')
142 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
143 plt.ylabel('Цена закрытия', fontsize=14, horizontalalignment='center')
144 plt.legend()
145 plt.show()
146
147 plt.figure(figsize=(10, 6))
148 plt.plot(yuan_real_stock_price1['close'], label='Реальный курс акций')
149 plt.plot(yuan_predicted_stock_price1['close'], label='Предсказанный курс
    акций')
150 plt.title('LSTM: Предсказание курса акций')
151 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
152 plt.ylabel('Цена закрытия', fontsize=14, horizontalalignment='center')
153 plt.legend()
154 plt.show()
155
156 yhat = yuan_data.loc['2021-06-22':, 'close']
157 evaluation_metric(finalpredicted_stock_price['close'], yhat)

```

#### Листинг 4: XGBoost модуль

```

1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from utils import *
5 from model import walk_forward_validation
6
7 data = pd.read_csv('./601988.SH.csv')
8 data.index = pd.to_datetime(data['trade_date'], format='%Y%m%d')
9 data = data.loc[:, ['open', 'high', 'low', 'close', 'vol', 'amount']]
10 # data = pd.DataFrame(data, dtype=np.float64)

```

```

11 close = data.pop('close')
12 data.insert(5, 'close', close)
13 data1 = data.iloc[3501:, 5]
14 residuals = pd.read_csv('./ARIMA_residuals1.csv')
15 residuals.index = pd.to_datetime(residuals['trade_date'])
16 residuals.pop('trade_date')
17 merge_data = pd.merge(data, residuals, on='trade_date')
18 #merge_data = merge_data.drop(labels='2007-01-04', axis=0)
19 time = pd.Series(data.index[3501:])
20
21 Lt = pd.read_csv('./ARIMA.csv')
22 Lt = Lt.drop('trade_date', axis=1)
23 Lt = np.array(Lt)
24 Lt = Lt.flatten().tolist()
25
26 train, test = prepare_data(merge_data, n_test=180, n_in=6, n_out=1)
27
28 y, yhat = walk_forward_validation(train, test)
29 plt.figure(figsize=(10, 6))
30 plt.plot(time, y, label='Остаток')
31 plt.plot(time, yhat, label='Предсказанный остаток')
32 plt.title('ARIMA+XGBoost: Предсказанный остаток')
33 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
34 plt.ylabel('Остаток', fontsize=14, horizontalalignment='center')
35 plt.legend()
36 plt.show()
37
38 finalpredicted_stock_price = [i + j for i, j in zip(Lt, yhat)]
39 #print('final', finalpredicted_stock_price)
40 evaluation_metric(data1, finalpredicted_stock_price)
41 plt.figure(figsize=(10, 6))
42 plt.plot(time, data1, label='Реальный курс акций')
43 plt.plot(time, finalpredicted_stock_price, label='Предсказанный курс акций')
44 plt.title('ARIMA+XGBoost: Предсказание курса акций')
45 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
46 plt.ylabel('Цена закрытия', fontsize=14, horizontalalignment='center')
47 plt.legend()
48 plt.show()

```

Листинг 5: Модуль пользовательского приложения

```

1 from keras.optimizers import Adam
2 import matplotlib.pyplot as plt
3 import pandas as pd
4 import numpy as np
5 from sklearn import metrics
6 from utils import *
7 from model import *
8
9 data1 = pd.read_csv("./601988.SH.csv")
10 data1.index = pd.to_datetime(data1['trade_date'], format='%Y%m%d')
11 #data1 = data1.drop(['ts_code', 'trade_date', 'turnover_rate', 'volume_ratio',
12                    'pb', 'total_share', 'float_share', 'free_share'], axis=1)
12 data1 = data1.loc[:, ['open', 'high', 'low', 'close', 'vol', 'amount']]
13 data_yuan = data1
14 residuals = pd.read_csv('./ARIMA_residuals1.csv')
15 residuals.index = pd.to_datetime(residuals['trade_date'])
16 residuals.pop('trade_date')
17 data1 = pd.merge(data1, residuals, on='trade_date')
18 data = data1.iloc[1:3500, :]
19 data2 = data1.iloc[3500:, :]
20
21 TIME_STEPS = 20
22
23 data, normalize = NormalizeMult(data)
24 print('#', normalize)
25 pollution_data = data[:, 3].reshape(len(data), 1)
26
27 train_X, _ = create_dataset(data, TIME_STEPS)
28 _, train_Y = create_dataset(pollution_data, TIME_STEPS)
29
30 print(train_X.shape, train_Y.shape)
31
32 m = attention_model(INPUT_DIMS=7)
33 m.summary()
34 adam = Adam(learning_rate=0.01)
35 m.compile(optimizer=adam, loss='mse')
36 history = m.fit([train_X], train_Y, epochs=50, batch_size=32,
37                validation_split=0.1)
37 m.save("./stock_model.h5")
38 np.save("stock_normalize.npy", normalize)

```

```

39
40 plt.plot(history.history['loss'], label='Потери на обучающем наборе')
41 plt.plot(history.history['val_loss'], label='Потери на проверочном наборе')
42 plt.title('Потери на обучающем и проверочном наборе')
43 plt.legend()
44 plt.show()
45
46 # normalize = np.load("normalize.npy")
47 # loadmodelname = "model.h5"
48
49 class Config:
50     def __init__(self):
51         self.dimname = 'close'
52
53 config = Config()
54 name = config.dimname
55 # normalize = np.load("normalize.npy")
56 y_hat, y_test = PredictWithData(data2, data_yuan, name, 'stock_model.h5', 7)
57 y_hat = np.array(y_hat, dtype='float64')
58 y_test = np.array(y_test, dtype='float64')
59 evaluation_metric(y_test, y_hat)
60 time = pd.Series(data1.index[3499:])
61 plt.plot(time, y_test, label='Курс акций')
62 plt.plot(time, y_hat, label='Предсказанный курс акций')
63 plt.title('Предсказание гибридной модели')
64 plt.xlabel('Дата', fontsize=12, verticalalignment='top')
65 plt.ylabel('Цена', fontsize=14, horizontalalignment='center')
66 plt.legend()
67 plt.show()

```