

Санкт–Петербургский государственный университет

*Сардарян Армен Саргисович*

Выпускная квалификационная работа  
*Прогнозирование фондовых индексов с  
использованием методов машинного обучения*

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»  
Профиль «Системный анализ, исследование операций и управление»

Научный руководитель:

доцент, кафедра теории систем  
управления электрофизической аппаратурой,  
к.ф. - м.н. Козынченко Владимир Александрович

Рецензент:

профессор, кафедра компьютерного  
моделирования и многопроцессорных систем,  
д.ф. - м.н. Андрианов Сергей Николаевич

Санкт-Петербург

2021 г.

# Содержание

|                                                                 |    |
|-----------------------------------------------------------------|----|
| <b>Введение</b> . . . . .                                       | 3  |
| <b>Постановка задачи</b> . . . . .                              | 6  |
| <b>Обзор литературы</b> . . . . .                               | 8  |
| <b>Глава 1. Авторегрессионный подход</b> . . . . .              | 10 |
| 1.1. Модель авторегрессионного скользящего среднего . . . . .   | 10 |
| 1.2. Стационарность временного ряда . . . . .                   | 12 |
| 1.3. Выбор оптимального порядка модели . . . . .                | 15 |
| 1.4. Оценка параметров модели . . . . .                         | 16 |
| 1.5. Проверка остатков модели . . . . .                         | 16 |
| <b>Глава 2. Нейросетевой подход</b> . . . . .                   | 17 |
| 2.1. Свёрточные нейронные сети . . . . .                        | 17 |
| 2.2. Перевод временного ряда в серию изображений . . . . .      | 21 |
| 2.2.1 Предобработка и разметка данных . . . . .                 | 22 |
| 2.2.2 Gramian Angular Field . . . . .                           | 23 |
| 2.2.3 Moving Average Mapping . . . . .                          | 25 |
| <b>Глава 3. Программная реализация и эксперименты</b> . . . . . | 27 |
| 3.1. Метрики . . . . .                                          | 28 |
| 3.2. Особенности реализации . . . . .                           | 29 |
| 3.3. Реализация авторегрессионного подхода . . . . .            | 30 |
| 3.4. Реализация нейросетевого подхода . . . . .                 | 32 |
| 3.4.1 Эксперименты с одноканальными моделями . . . . .          | 34 |
| 3.4.2 Эксперименты с многоканальными моделями . . . . .         | 36 |
| 3.5. Сравнение моделей . . . . .                                | 38 |
| <b>Выводы</b> . . . . .                                         | 42 |
| <b>Заключение</b> . . . . .                                     | 44 |
| <b>Список литературы</b> . . . . .                              | 45 |

## Введение

В настоящее время к прогнозированию фондовых индексов привлечено большое внимание со стороны профессиональных трейдеров, инвесторов и исследователей. Фондовый индекс — это показатель, вычисляемый определённым образом на основе значений котировок группы ценных бумаг. Фондовые индексы, в зависимости от своего назначения, позволяют оценить состояние рынка ценных бумаг в целом или состояние какой-либо отрасли. Успешное прогнозирование будущих изменений индексов и соответствующие прогнозам принятия решений могут принести ощутимую прибыль.

Прогнозирование фондовых индексов — это попытка предсказать вероятное изменение их значений в будущем. Даже прогнозирование с точностью немногим выше случайного угадывания вызывает интерес, так как эта небольшая разница в долгосрочной перспективе может принести прибыль трейдерам или инвесторам. Целью разработок новых методов является повышение этой точности и, как следствие, увеличение прибыли.

Существует множество методов прогнозирования фондовых индексов и акций, среди которых можно выделить 2 большие группы: фундаментальный анализ и технический анализ.

В основе фундаментального анализа лежит сам объект прогнозирования (фондовый индекс, акция компании): его финансовые показатели, макроэкономические и политические факторы, влияющие на объект. Фундаментальный анализ — всеобъемлющий инструмент, он проводится на разных уровнях: от геополитических событий, состояния мировой экономики до отдельной отрасли или компании. На прогноз могут влиять, например, цены на нефть, темпы роста ВВП страны, уровень инфляции, курс валюты, ставка Центрального Банка. Фундаментальный анализ хорошо подходит для долгосрочных стратегий, поэтому его активно используют инвесторы, которые отбирают наиболее выгодные компании для составления своего инвестиционного портфеля.

Для позиционного трейдинга, то есть для краткосрочных стратегий, лучше подходит технический анализ. Технический анализ использует толь-

ко ценовой ряд, прогноз составляется на основе закономерностей изменения цен в прошлом в похожих обстоятельствах. Инструменты технического анализа — графики (тиковые, японские свечи, пункт-цифровые), паттерны — устойчивые повторяющиеся сочетания цен (например, «голова-плечи», «вымпел», «двойная вершина») [1], а также технические индикаторы — функции от значений ценового ряда (скользящая средняя, индекс относительной силы, MACD, линии Болинджера, японские свечи) [2]. В настоящее время также успешно применяются авторегрессионные модели и современные методы глубокого обучения.

Среди авторегрессионных моделей наиболее популярна модель авторегрессионного скользящего среднего (ARMA), предложенная в 1951 году Питером Уайтли в работе «Hypothesis Testing in Time Series Analysis» [3] и впоследствии доработанная и популяризованная Джорджем Боксом и Гвиллимом Дженкинсом в 1970-х.

С середины 2000-х годов глубокое обучение набирает популярность как эффективный способ решения многих задач в различных сферах. Глубокие нейронные сети достигли значительного успеха в решении задач компьютерного зрения и распознавания речи в 2011-2012 годах, после чего стали активно использоваться повсеместно.

Успех глубоких нейронных сетей подтолкнул исследователей к идее их применения к решению задач технического анализа, ведь вполне возможно, что нейронные сети смогут заметить такие закономерности, которые остаются незамеченными человеком. Свёрточные нейронные сети должны получать двумерные данные в качестве входных, поэтому их неэффективно применять непосредственно к временным рядам. В 2015 году появилась идея кодирования временных рядов в виде изображений и применения к ним свёрточных нейронных сетей для «визуального» распознавания и изучения закономерностей [4].

В 2015-2016 годах были разработаны разные типы кодирования временных рядов в изображения, такие как Gramian Angular Field [4], Moving Average Mapping, Double Moving Average Mapping [5] и другие. Свёрточная нейронная сеть обучалась на изображениях, полученных с помощью кодировщиков. Сеть находила закономерности между изображением и её

классом. Данный подход можно использовать в прогнозировании фондовых индексов: обучившись на достаточном количестве изображений, сеть сможет предсказывать к какому классу относится новое изображение.

Экспериментируя с обработкой временного ряда, методами кодирования, в том числе их комбинированием, а также с архитектурой свёрточной нейронной сети можно добиться относительно высокого качества прогнозирования.

## Постановка задачи

Целью данной работы является построение моделей прогнозирования изменения фондовых индексов с помощью методов машинного обучения и их сравнение. В работе рассматриваются 2 подхода к решению задачи: авторегрессионный и нейросетевой.

Для достижения поставленной цели необходимо решить ряд задач:

1. поиск, предобработка и разметка данных
2. реализация авторегрессионного подхода:
  - (a) проверка стационарности временного ряда
  - (b) выбор оптимального порядка модели
  - (c) построение модели авторегрессионного скользящего среднего
  - (d) проверка остатков модели
3. реализация нейросетевого подхода:
  - (a) перевод временного ряда в серию изображений с использованием методов кодирования Gramian Angular Field, Moving Average Mapping и Double Moving Average Mapping
  - (b) построение свёрточной нейронной сети
  - (c) эксперименты с одноканальными и многоканальными моделями
4. сравнение построенных моделей.

Задача прогнозирования в данной работе формулируется следующим образом. Пусть  $X = \{X_1, X_2, \dots, X_n\}$  – исходный ценовой ряд некоторого фондового индекса. Отрезок  $\{X_{i-k+1}, X_{i-k+2}, \dots, X_i\}$  ( $i \geq k$ ) временного ряда  $X$  отнесём к классу Up, если значение  $X_{i+r}$  удовлетворяет неравенству:  $\frac{X_{i+r}-X_i}{X_i} > l_2$ , к классу Down, если  $\frac{X_{i+r}-X_i}{X_i} < l_1$ , к классу Flat – в противном случае. Величины  $k, r, l_1, l_2$  – задаваемые параметры. Таким образом, задача заключается в классификации каждого нового отрезка к одному из  $s = 3$  классов.

В качестве тестовых данных были взяты фондовые индексы США и Японии — S&P 500 и Nikkei — за период с 1 января 1992 года по 17 февраля 2021 года со значениями на момент открытия и закрытия торгов по дням. В корзину S&P 500 включены 505 торгуемых на фондовых рынках США публичных компаний с наибольшей капитализацией. В корзину Nikkei включены 225 крупнейших компаний, торгуемых на Токийской фондовой бирже.

## Обзор литературы

Вопросу прогнозирования фондовых индексов посвящено большое количество литературы.

В труде Владимира Меладзе «Курс технического анализа» [2] рассмотрены основные теории, инструменты и индикаторы технического анализа, а также кратко описан принцип работы автоматических торговых систем. В главе 3 труда «Всё об индикаторах рынка» Майкла Синсиэра [6] описаны основные технические индикаторы и примеры их применения на практике. Графический подход подробнее описан в работе «Технический анализ: графический подход» Лозговцева Ю., Сафина В., Федорова А. [1].

В книге «Time Series Analysis: Forecasting and Control» Box G., Jenkins G., Reinsel G., Ljung G. [7] подробно рассматриваются различные модели прогнозирования временных рядов, в том числе описаны авторегрессионные модели, способы проверки временного ряда на стационарность, выбор оптимального порядка и процесс обучения моделей. Тест Дики-Фуллера для проверки временного ряда на стационарность подробно описан в статье «Distribution of the Estimates for Autoregressive Time Series with a Unit Root» Dickey D., Fuller W. [8].

В работе «Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks», Wang Zh., Oates T. [4] описаны 2 способа кодирования временных рядов в изображения: Gramian Angular Field (GAF) и Markov Transition Field (MTF), а также приведено математическое обоснование использования данных методов. Во второй части работы рассматривается применение плиточной свёрточной нейронной сети (Tiled CNN) для классификации представлений GAF и MTF на 12 разных наборах данных. Кроме того, в данной работе ставится эксперимент по комбинированию этих 2 методов кодирования.

Другие методы кодирования, а именно Moving Average Mapping и Double Moving Average Mapping, описаны в работе «Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks», Chen J., Chen W., Huang C., Huang S., Chen A. [5]. Данные методы используют понятие скользящего среднего для построения изображений. Результаты работы моделей



с этими методами кодирования сравниваются с результатом модели с кодировщиком GAF. На некоторых тестовых данных качество методов МАМ или ДМАМ превосходит GAF.

Исследователи из университета Кальяри (Италия) в своей работе «Deep Learning and Time Series-to-Image Encoding for Financial Forecasting» [9] предлагают использовать ансамбль одинаковых свёрточных нейронных сетей для классификации изображений, полученных методом GAF. Авторам удалось достичь достаточно высоких результатов на тестовых данных по сравнению с другими моделями.

В книге Орельена Жерона «Прикладное машинное обучение с помощью Scikit-Learn и Tensorflow: концепции, инструменты и техники для создания интеллектуальных систем» [10] разобраны алгоритмы машинного обучения и их реализация с помощью библиотек языка программирования Python, а также описаны основные способы проверки качества моделей, что нужно для проведения сравнительного анализа.

# Глава 1. Авторегрессионный подход

Пусть имеется ценовой ряд  $X = \{X_1, X_2, \dots, X_n\}$ . Перейдем от абсолютных значений к относительным. Рассмотрим изменение цен с лагом  $r$ . Получим новый ряд, состоящий из  $n - r$  элементов:

$$Y_1 = \frac{X_{r+1} - X_1}{X_1}, Y_2 = \frac{X_{r+2} - X_2}{X_2}, \dots, Y_{n-r} = \frac{X_n - X_{n-r}}{X_{n-r}} \quad (1)$$

Авторегрессионный подход к прогнозированию изменения фондового индекса через  $r$  дней заключается в применении классической модели авторегрессионного скользящего среднего (ARMA) с оптимальными параметрами  $p$  и  $q$  к полученному временному ряду  $Y = \{Y_1, Y_2, \dots, Y_T\}$ , где  $T = n - r$  и последующей классификацией спрогнозированных значений.

## 1.1 Модель авторегрессионного скользящего среднего

Моделью авторегрессионного скользящего среднего ARMA( $p, q$ ) называется процесс генерации временного ряда, являющийся комбинацией процессов авторегрессии AR( $p$ ) и скользящего среднего MA( $q$ ). ARMA используется для прогнозирования стационарных временных рядов.

Авторегрессионная модель AR( $p$ ) — это модель временного ряда, в котором текущее значение ряда линейно зависит от  $p$  предыдущих значений этого же ряда. Авторегрессионный процесс порядка  $p$  определяется следующим образом:

$$Y_t = c + \sum_{i=1}^p \phi_i Y_{t-i} + \epsilon_t \quad (2)$$

где  $c$  — константа,  $\phi_1, \phi_2, \dots, \phi_p$  — коэффициенты авторегрессии (параметры модели),  $\epsilon_t$  — некоторая случайная компонента в текущий момент времени.

Модель скользящего среднего MA( $q$ ) — это модель временного ряда, в котором текущее значение ряда линейно зависит от случайных компонент в текущий и  $q$  предыдущих моментах времени. Процесс скользящего

среднего порядка  $q$  определяется следующим образом:

$$Y_t = \mu + \epsilon_t + \sum_{i=1}^q \theta_i \epsilon_{t-i} \quad (3)$$

где  $\mu$  — математическое ожидание ряда,  $\theta_1, \theta_2, \dots, \theta_q$  — параметры модели,  $\epsilon_t, \epsilon_{t-1}, \dots, \epsilon_{t-q}$  — случайные компоненты в текущий и  $q$  предыдущих моментах времени.

Случайные компоненты  $\epsilon_t$  обычно считаются независимыми одинаково распределёнными случайными величинами, взятыми из нормального распределения с нулевым математическим ожиданием и постоянной дисперсией:  $\epsilon_t \sim N(0, \sigma^2)$ , где  $\sigma^2$  — дисперсия. Стационарный ряд, представляющий из себя набор таких случайных величин, называется белым шумом.

Модель авторегрессионного скользящего среднего ARMA( $p, q$ ) состоит из  $p$  членов авторегрессии и  $q$  членов скользящего среднего. Процесс ARMA( $p, q$ ) является комбинацией процессов AR( $p$ ) и MA( $q$ ), т. е. текущее значение временного ряда  $Y_t$  определяется как сумма правых частей уравнений 2 и 3:

$$Y_t = c + \epsilon_t + \sum_{i=1}^p \phi_i Y_{t-i} + \sum_{i=1}^q \theta_i \epsilon_{t-i} \quad (4)$$

где  $c$  — константа,  $\{Y_i\}_{i=t-p}^{t-1}$  —  $p$  предыдущих значений временного ряда,  $\{\epsilon_i\}_{i=t-q}^t$  — белый шум,  $\phi_i, \theta_i$  — действительные числа, параметры модели.

Константа  $c$  в формуле 4 может интерпретироваться как математическое ожидание временного ряда в случае отсутствия параметров авторегрессии (т.е. при  $\phi_i = 0 \forall i \in [t-p; t-1]$ ) или как свободный член в противном случае.

Модель ARMA( $p, q$ ) более сложная по структуре, чем модели AR( $p$ ), MA( $q$ ), однако обычно она имеет меньшее количество параметров, что является её преимуществом. Недостатком модели является то, что она применима только к стационарным временным рядам. Для нестационарных временных рядов существует модификация модели ARMA( $p, q$ ) — интегрированная авторегрессионная модель скользящего среднего ARIMA( $p,$

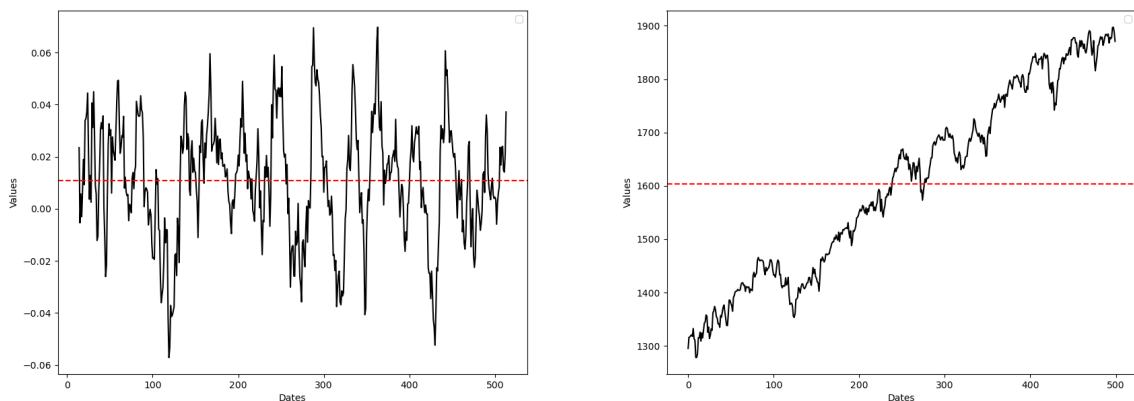
$q, d$ ). Ряд описывается моделью  $ARIMA(p, q, d)$ , если ряд его разностей порядка  $d$  описывается моделью  $ARMA(p, q)$ .

На практике изначально неизвестны характеристики временного ряда, поэтому для построения качественной модели предлагается следующий подход, называемый методологией Бокса—Дженкинса:

1. идентификация модели: проверить ряд на стационарность — определить порядок интегрируемости  $d$ , определить порядки авторегрессии и скользящего среднего  $(p, q)$
2. обучение модели: оценивание параметров  $(\theta_i, \phi_i, c)$
3. проверка остатков модели: остатки должны быть белым шумом.

## 1.2 Стационарность временного ряда

Согласно теореме Волда, любой стационарный ряд может быть аппроксимирован моделью  $ARMA(p, q)$  с достаточной степенью точности. Однако, как известно, существует модель и для нестационарных рядов —  $ARIMA(p, q, d)$ . Для того, чтобы определиться с моделью, нужно проверить ряд на стационарность.



**Рис. 1:** Стационарный (слева) и нестационарный (справа) временные ряды

Временной ряд называется стационарным, если его математическое ожидание и дисперсия постоянны и конечны. Постоянство математического ожидания означает, что ряд колеблется вокруг некоторого среднего, а

постоянство дисперсии требует одинаковый разброс значений ряда от среднего в зависимости от времени. Очевидно, что стационарный временной ряд не имеет тренда. На рис. 1 приведены примеры графиков стационарного и нестационарного временных рядов соответственно.

Нестационарность временного ряда характеризуется наличием единичных корней. Говорят, что временной ряд имеет единичный корень, если его первые разности  $\Delta Y_t = Y_t - Y_{t-1}$  образуют стационарный ряд.

Для определения стационарности временного ряда используются тесты на единичные корни, среди которых наиболее известным является тест Дики–Фуллера. Тест Дики-Фуллера был предложен Дэвидом Дики и Уэйном Фуллером в 1979 году в работе [8].

Рассмотрим авторегрессионную модель первого порядка AR(1):

$$Y_t = aY_{t-1} + \epsilon_t \quad (5)$$

Если  $a = 1$ , то временной ряд имеет единичный корень, и, следовательно, он нестационарен. В этом случае говорят, что ряд является интегрированным временным рядом первого порядка.

Запишем уравнение авторегрессии 5 через первые разности временного ряда:  $\Delta Y_t = bY_{t-1} + \epsilon_t$ , где  $b = a - 1$ . В качестве нулевой гипотезы теста Дики–Фуллера рассматривается гипотеза о равенстве нулю коэффициента  $b$ , в качестве альтернативной —  $b < 0$ :

$$H_0 : b = 0$$

$$H_1 : b < 0$$

В [8] показано, что статистика теста (DF-статистика) не подчиняется распределению Стьюдента. Распределение DF-статистики называется распределением Дики–Фуллера.

Критические значения зависят от вида уравнения авторегрессии. Возможны 3 случая:

1. уравнение авторегрессии без константы и тренда:

$$\Delta Y_t = bY_{t-1} + \epsilon_t$$

2. уравнение авторегрессии с константой, но без тренда:

$$\Delta Y_t = c + bY_{t-1} + \epsilon_t$$

3. уравнение авторегрессии с константой и линейным трендом:

$$\Delta Y_t = c_0 + c_1t + bY_{t-1} + \epsilon_t$$

Для каждого из трёх типов уравнения авторегрессии существуют свои критические значения DF-статистики. Если значение DF-статистики лежит левее критического значения при данном уровне значимости, то нулевая гипотеза о равенстве нулю коэффициента  $b$  отвергается, и, следовательно, отклоняется гипотеза о единичном корне, т. е. ряд признаётся стационарным. В противном случае гипотеза не отвергается и ряд может быть нестационарным.

Для авторегрессионной модели более высокого порядка необходимо воспользоваться расширенным тестом Дики–Фуллера, который заключается в добавлении лагов первых разностей в уравнение авторегрессии. В этом случае модель будет иметь следующий вид (для третьего типа уравнения):

$$\Delta Y_t = c_0 + c_1t + bY_{t-1} + \sum_{i=1}^{p-1} \gamma_i \Delta Y_{t-i} + \epsilon_t$$

где  $c_0$  — константа,  $c_1$  — коэффициент при линейном тренде,  $p$  — порядок модели авторегрессии  $AR(p)$ . В случае  $c_1 = 0$  получаем второй тип уравнения авторегрессии, а в случае  $c_1 = c_0 = 0$  — первый.

Процедура проведения теста, а также критические значения не меняются: нулевая гипотеза также гипотеза о равенстве нулю коэффициента  $b$ , который теперь учитывает и добавленные лаги первых разностей, т. е.

$b = \sum_{i=1}^p a_i - 1$ , где  $a_i$  коэффициенты при  $p$  предыдущих значениях ряда.

Если ряд нестационарен, но стационарен ряд его разностей порядка  $k$ , то ряд имеет  $k$ -ый порядок интегрируемости и, следовательно, для него подходит модель  $ARIMA(p, q, d)$  с  $d = k$ . Тест Дики–Фуллера проверяет наличие лишь одного единичного корня. Для установления порядка интегрируемости нужно провести тест Дики–Фуллера для ряда разностей порядка  $k_1$  (в реальных экономических задачах ограничиваются порядком 3). Если гипотеза о наличии единичного корня отвергается, то проводится такой же тест для ряда разностей порядка  $k_2 = k_1 - 1$  и т. д. Если гипотеза для ряда разностей порядка  $k_i$  не отвергается, то исходный ряд имеет  $k_i + 1$  единичных корней, и, следовательно, для него используется модель  $ARIMA(p, q, d)$  с  $d = k_i + 1$ .

### 1.3 Выбор оптимального порядка модели

Выбор оптимального порядка модели  $ARMA(p, q)$  основан на информационном критерии Акаике. Для выбора оптимального порядка нужно перебрать все значения  $p$  и  $q$  (обычно ограничиваются 3-5 порядком), построить модели-кандидаты  $ARMA(p, q)$  и для каждой из них определить значение  $AIC$  по формуле:

$$AIC = -2\log(L) + 2(p + q + k + 1)$$

где  $L$  — логарифм правдоподобия модели,  $k = 1$  при  $c \neq 0$  и  $k = 0$  при  $c = 0$ ,  $c$  — константа модели. Чаще применяется скорректированный  $AIC$ , который имеет следующий вид:

$$AIC_c = AIC + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2}$$

где  $T$  — длина временного ряда.

Модель с наименьшим значением  $AIC$  ( $AIC_c$ ) и будет оптимальной.

## 1.4 Оценка параметров модели

Параметры модели (коэффициенты  $\phi_i, \theta_i$ , а также константа  $c$ ) могут быть оценены как с помощью метода наименьших квадратов, так и методом максимального правдоподобия.

В методе наименьших квадратов минимизируется остаточная сумма квадратов  $\sum_{i=1}^T e_i^2$ , где  $e_i$  — ошибка  $i$ -го значения ряда. Ошибка — это разница между фактическим и оценённым значениями временного ряда.

В методе максимального правдоподобия максимизируется логарифмическая функция правдоподобия модели.

## 1.5 Проверка остатков модели

Прежде чем воспользоваться построенной моделью для прогноза, нужно проверить её остатки (ошибки) на соответствие белому шуму. В случае, если остатки не являются белым шумом, нужно перестроить модель с другими значениями  $p$  и  $q$ .

Ряд можно считать белым шумом, если между его значениями отсутствует взаимосвязь, взятая со сдвигом во времени. Для проверки необходимо воспользоваться критерием Льюнга–Бокса для выявления автокорреляции временных рядов. Выдвигаются конкурирующие гипотезы:

$H_0$ : значения временного ряда статистически независимы,

$H_1$ : значения временного ряда не являются статистически независимыми.

Далее вычисляется  $Q$ -статистика по формуле:

$$Q = T(T + 2) \sum_{i=1}^m \frac{\rho_i^2}{T - i}$$

где  $T$  — длина временного ряда,  $\rho_i$  — автокорреляция  $i$ -го порядка,  $m$  — количество проверяемых лагов. Если при заданном уровне значимости  $\alpha$   $Q > \chi_{1-\alpha, m}^2$ , где  $\chi_{1-\alpha, m}^2$  — квантиль распределения хи-квадрат с  $m$  степенями свободы, то нулевая гипотеза отвергается и признаётся наличие автокорреляции до  $m$ -го порядка во временном ряде. В противном случае считается, что ряд остатков модели является белым шумом.



## Глава 2. Нейросетевой подход

Нейросетевой подход к прогнозированию изменения фондового индекса через  $r$  дней заключается в применении свёрточной нейронной сети к закодированному отрезку временного ряда  $X$ . Результаты прогнозирования могут зависеть от многих обстоятельств: архитектуры нейронной сети, её гиперпараметров, методов кодирования временного ряда в изображения, а также предобработки данных.

### 2.1 Свёрточные нейронные сети

Свёрточная нейронная сеть (Convolutional Neural Network, CNN) — это многослойная нейронная сеть прямого распространения, основная идея которой заключается в чередовании свёрточных и субдискретизирующих (пулинговых) слоёв.

Первая подобная архитектура — LeNet — была предложена Яном Лекуном в 1988 году в работе [11]. Свёрточные нейронные сети долгое время были непопулярными из-за отсутствия необходимого для вычислений оборудования. Однако с увеличением вычислительной мощности компьютеров интерес к свёрточным нейронным сетям возрос. В первой половине 2010-х годов, после значительных успехов нескольких архитектур (AlexNet, ResNet, VGG, GoogLeNet) в решении задачи распознавания образов, свёрточные нейронные сети получили всеобщее признание и стали использоваться повсеместно.

Свёрточный слой — это основной блок свёрточной нейронной сети. Нейроны в таких слоях, в отличие от полносвязных, связаны не с каждым нейроном в предыдущем слое, а только с нейронами в ограниченном прямоугольнике, который называется рецептивным полем. Веса нейронов представлены в виде матрицы той же размерности, что и рецептивное поле. Такая матрица называется ядром свёртки или фильтром. Значения фильтров являются обучаемыми параметрами сети.

Основу вычислений в свёрточном слое составляет свёртка — операция над двумя матрицами  $A_{n_1, n_2}$  и  $B_{m_1, m_2}$  ( $m_1 \geq n_1, m_2 \geq n_2$ ), результатом которой является матрица  $C_{m_1 - n_1 + 1, m_2 - n_2 + 1}$ . Каждый элемент  $C_{x, y}$  матрицы

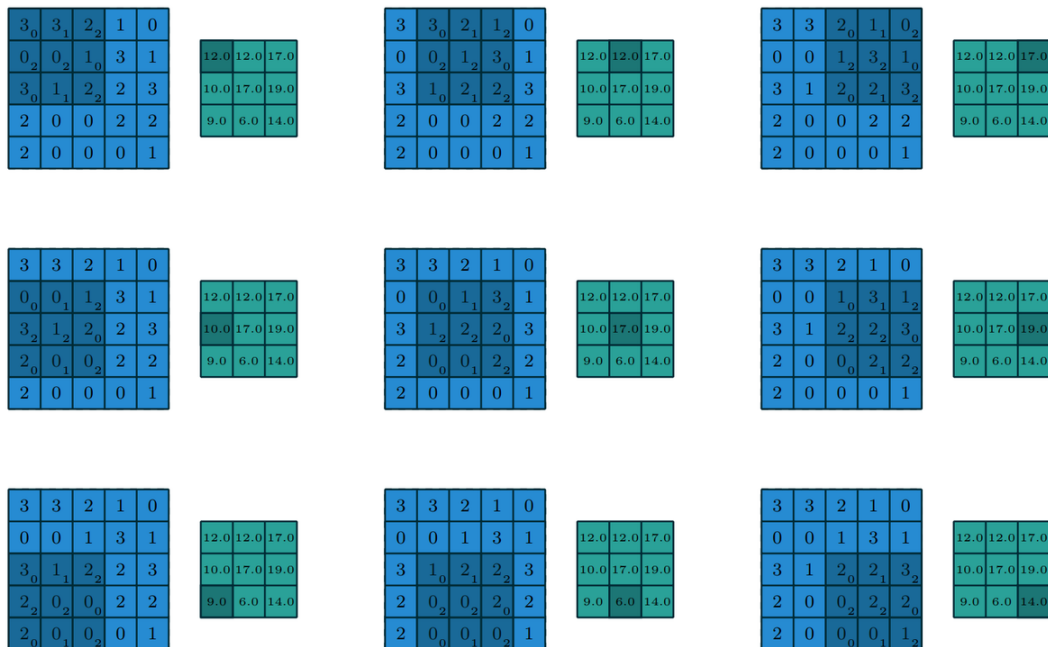
$C$  вычисляется следующим образом:

$$C_{x,y} = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} A_{i,j} B_{x+i,y+j} \quad (6)$$

Пример операции свёртки показан на рис. 2.

Свёрточный слой представляет из себя применение операции свёртки к матрицам фильтра и входных данных. В качестве входных данных первый слой обычно получает изображение, следовательно, признаками являются пиксели. Последующие слои получают на вход карту признаков — выходы нейронов предыдущего слоя, на котором в результате операции свёртки выделились некоторые высокоуровневые признаки (например, линии, углы, фигуры).

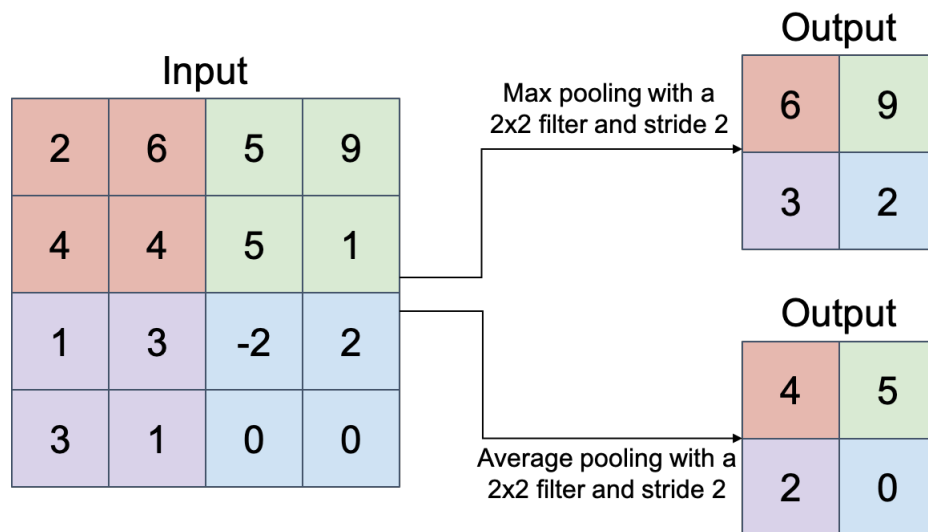
Свёрточный слой выдаёт карту признаков, которая выделяет области изображения, наиболее схожие с применяемым фильтром. В процессе обучения свёрточная нейронная сеть находит лучшие фильтры для решаемой задачи и учится комбинировать их в более сложные образы для выделения высокоуровневых признаков.



**Рис. 2:** Операция свёртки двух матриц размером  $5 \times 5$  и  $3 \times 3$  со страйдом 1 [12]

Для вычисления каждого нового значения слоя нужно применить формулу 6 к соответствующей области входной карты признаков (рецептивному полю) и фильтру. Смещение между двумя соседними рецептивными полями называется страйдом. На рис. 2 страйд равен 1 в обоих направлениях. В случае, если страйд  $s$  больше 1, то размер итоговой матрицы будет  $(\frac{m_1-n_1}{s} + 1) \times (\frac{m_2-n_2}{s} + 1)$ . Входную карту признаков можно дополнить нулями слева, справа, сверху и снизу. Эта операция называется дополнением нулями (zero padding).

Карты признаков также называются каналами. Входные данные могут быть как одноканальными, так и многоканальными. Например, для полутоновых изображений (в градациях серого) используется только один канал, а для цветных — три, каждый из которых соответствует одному цвету: красному, зелёному и синему. Однако, в общем случае, количество каналов не ограничено.



**Рис. 3:** Пулинговый слой с функциями максимума и среднего и страйдом 2 [13]

Для понижения размерности карт признаков предыдущего слоя используются субдискретизирующие (пулинговые) слои. Пулинговые слои позволяют сократить вычислительную нагрузку, расход памяти, а также уменьшить риск переобучения.

Как и в свёрточных слоях, каждый нейрон в пулинговом слое связан с выходами ограниченного числа нейронов предыдущего слоя, кото-

рые расположены внутри рецептивного поля. Однако фильтр в пулинговом слое не имеет весов, а лишь представляет из себя применение некоторой функции к каждому рецептивному полю. Чаще всего используется функция максимума (max-pooling): на следующий слой передаётся только один максимальный элемент рецептивного поля. Также достаточно распространена функция среднего (average pooling): на следующий слой передаётся арифметическое среднее элементов рецептивного поля. Примеры пулинга по максимальному и среднему значениям показаны на рис. 3.

В типовой архитектуре свёрточной нейронной сети чередуются свёрточные и пулинговые слои, а в конце добавляются слой упрочения (flatten) и несколько полносвязных слоёв. Слой упрочения преобразует выходы с предыдущего слоя в вектора. Полносвязный слой выступает в роли классификатора, на вход которому подаются признаки, выделенные с помощью свёрточных слоёв. Его основной задачей является моделирование некоторой функции, которая оптимизируется в ходе обучения сети. Последний полносвязный слой выдаёт прогноз (например, оценочные вероятности классов).

Для обучения свёрточной нейронной сети чаще всего используется метод обратного распространения ошибки, который является модификацией классического метода градиентного спуска [14].

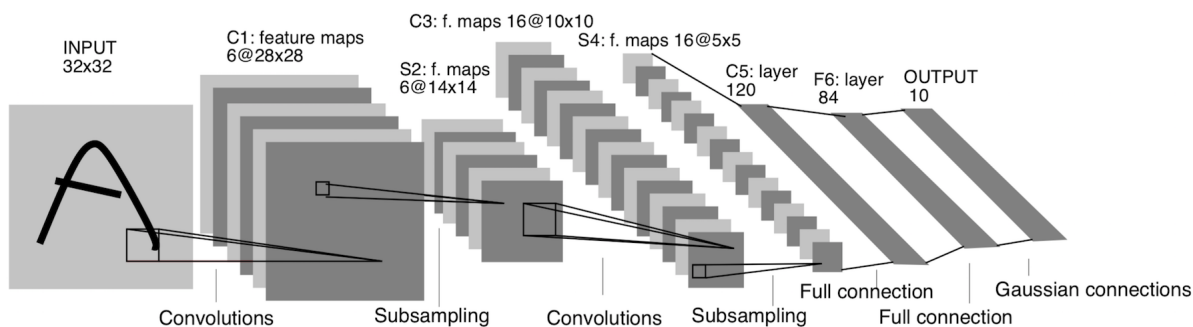


Рис. 4: Архитектура LeNet-5 [15]

Архитектура LeNet-5, созданная Яном Лекуном в 1998 году [15], является одной из самых известных архитектур свёрточных нейронных сетей. Эта сеть использовалась для распознавания рукописных цифр. LeNet-5 со-

стоит из 3 свёрточных слоёв с размером ядра  $5 \times 5$ , 2 пулинговых слоёв с функцией среднего и размером ядра  $2 \times 2$  и 2 полносвязных слоёв. Архитектура LeNet-5 показана на рис. 4.

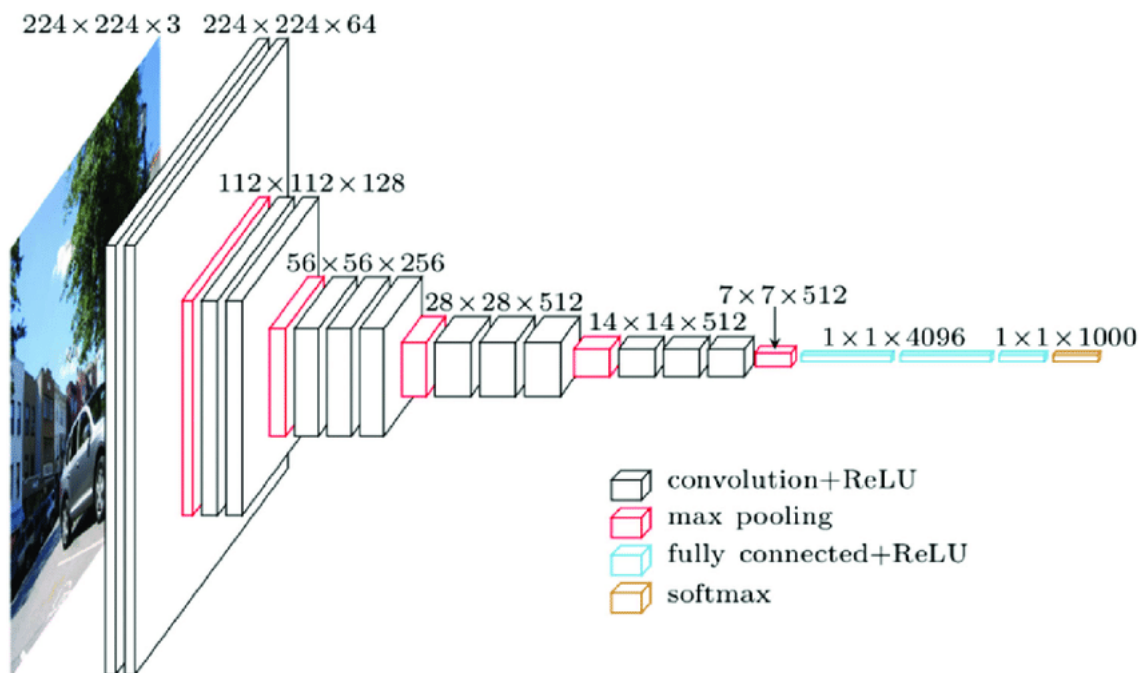


Рис. 5: Архитектура VGG-16 [16]

В первой половине 2010-х годов были созданы ещё несколько архитектур свёрточных нейронных сетей, одна из которых — VGG. VGG был предложен Кареном Симоньяном и Эндрю Зиссерманом в 2015 году в работе [17]. В VGG, в отличие от LeNet-5, пулинговые слои чередуются не с свёрточными, а с блоками из нескольких свёрточных слоёв. Архитектура VGG-16, то есть VGG с 16 свёрточными и 5 пулинговыми слоями с функцией максимума, представлена на рис. 5

## 2.2 Перевод временного ряда в серию изображений

Свёрточные нейронные сети как правило работают с двумерными данными в качестве входных. Поэтому для их использования временной ряд нужно перевести в двумерные матрицы, то есть в изображения (тепловые карты). Этот процесс называется кодированием временного ряда.

## 2.2.1 Предобработка и разметка данных

Для перевода временного ряда в серию изображений нужно представить его в виде набора малых временных рядов размера  $k$  с лагом  $l$ . Назовём такие ряды окнами.

Преобразуем исходный временной ряд  $X = \{X_1, X_2, \dots, X_n\}$  в набор  $m = \lceil \frac{n-k+1}{l} \rceil$  окон размера  $k$  с лагом  $l$ :

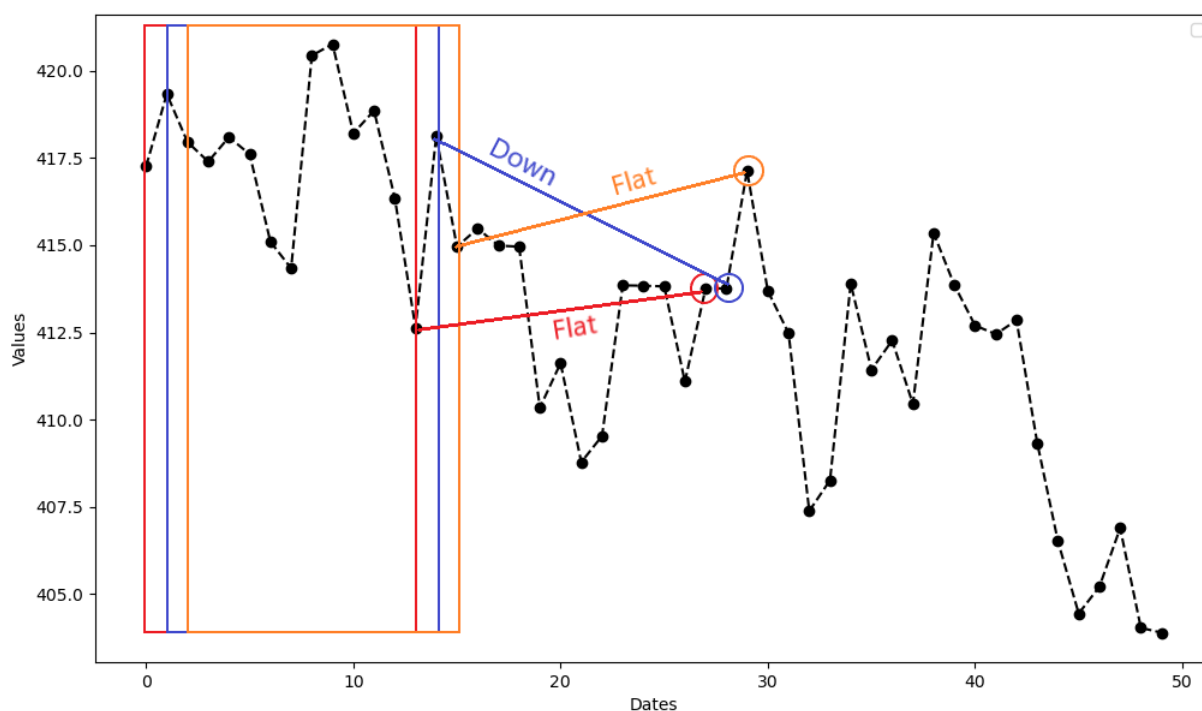
$$X^1 = \{X_1, X_2, \dots, X_k\}$$

$$X^{1+l} = \{X_{1+l}, X_{2+l}, \dots, X_{k+l}\}$$

...

$$X^p = \{X_p, X_{p+1}, \dots, X_{p+k-1}\}$$

где  $p = 1 + (m - 1)l$ .



**Рис. 6:** 3 окна размера  $k = 14$  и с лагом  $l = 1$  и соответствующие им метки ( $r = 14$ )

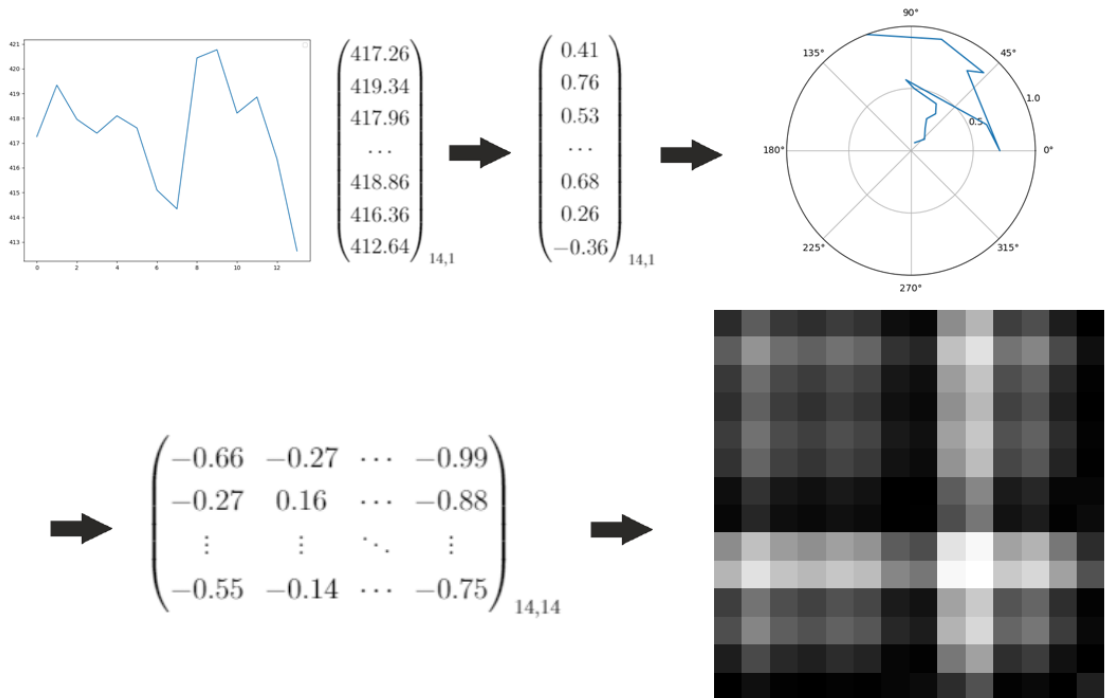
Сопоставим каждому окну метку — идентификатор одного из  $s$  классов, которые должна определить нейронная сеть. Если цель — спрогнозировать изменение фондового индекса через  $r$  дней, то метка класса будет

определяться по изменению  $(i + r)$ -ого значения ряда относительно  $i$ -ого, где  $i$  — это номер последнего значения соответствующего окна. Примеры размеченных окон показаны на рис. 6.

Очевидно, что количество размеченных окон будет на  $d = \lceil \frac{r}{l} \rceil$  меньше, то есть всего останется  $m - d$  окон, готовых к использованию. Но прежде нужно закодировать каждое окно в изображение.

## 2.2.2 Gramian Angular Field

Gramian Angular Field (GAF) — это способ кодирования временного ряда в изображение, предложенный Тимом Оатесом и Чжигуаном Вангом в 2015 году в статье [4]. Временной ряд переводится в полярную систему координат. На основе этих данных строится матрица  $G$ , каждый элемент которой является косинусом суммы соответствующих углов. По этой матрице строится тепловая карта, которая будет одним из изображений, подаваемых на вход свёрточной нейронной сети.



**Рис. 7:** Процесс кодирования временного ряда размера  $k = 14$  методом GAF

Рассмотрим временной ряд  $X = \{X_1, X_2, \dots, X_k\}$ . Матрица  $G$  для

этого временного ряда строится следующим образом:

1. Производится масштабирование каждого значения ряда в интервал  $[-1, 1]$  по формуле:

$$\bar{X}_i = \frac{2X_i - \max(X) - \min(X)}{\max(X) - \min(X)}$$

Масштабирование обеспечивает попадание полярного угла в промежуток  $[0, \pi]$ .

2. Полученный временной ряд  $\bar{X} = \{\bar{X}_1, \bar{X}_2, \dots, \bar{X}_k\}$  переводится в полярную систему координат по формулам:

$$\begin{cases} \phi_i = \arccos(\bar{X}_i), & \bar{X}_i \in \bar{X} \\ r_i = \frac{t_i}{N}, & t_i \in \mathbb{N} \end{cases} \quad (7)$$

где  $N$  – коэффициент регуляризации для диапазона полярного пространства (обычно устанавливается  $N = 1$ ),  $t_i$  – момент времени.

3. Матрица  $G$  вычисляется по формуле:

$$G = \begin{pmatrix} \cos(\phi_1 + \phi_1) & \cos(\phi_1 + \phi_2) & \cdots & \cos(\phi_1 + \phi_k) \\ \cos(\phi_2 + \phi_1) & \cos(\phi_2 + \phi_2) & \cdots & \cos(\phi_2 + \phi_k) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_k + \phi_1) & \cos(\phi_k + \phi_2) & \cdots & \cos(\phi_k + \phi_k) \end{pmatrix}$$

С учётом формулы косинуса суммы  $\cos(\phi_i + \phi_j) = \cos(\phi_i)\cos(\phi_j) - \sin(\phi_i)\sin(\phi_j) = \cos(\phi_i)\cos(\phi_j) - \sqrt{1 - \cos(\phi_i)^2}\sqrt{1 - \cos(\phi_j)^2}$  и первой формулы в системе 7 каждый элемент матрицы  $G$  можно переписать следующим образом:

$$\langle \bar{X}_i, \bar{X}_j \rangle \stackrel{\text{def}}{=} \bar{X}_i \bar{X}_j - \sqrt{1 - \bar{X}_i^2} \sqrt{1 - \bar{X}_j^2}$$



Таким образом, матрица  $G$  – это матрица Грама:

$$G = \begin{pmatrix} \langle \bar{X}_1, \bar{X}_1 \rangle & \langle \bar{X}_1, \bar{X}_2 \rangle & \cdots & \langle \bar{X}_1, \bar{X}_k \rangle \\ \langle \bar{X}_2, \bar{X}_1 \rangle & \langle \bar{X}_2, \bar{X}_2 \rangle & \cdots & \langle \bar{X}_2, \bar{X}_k \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \bar{X}_k, \bar{X}_1 \rangle & \langle \bar{X}_k, \bar{X}_2 \rangle & \cdots & \langle \bar{X}_k, \bar{X}_k \rangle \end{pmatrix}$$

Затем на основе матрицы  $G_{k,k}$  строится тепловая карта такой же размерности. Пример работы алгоритма GAF приведен на рис. 7.

Преобразования по формуле 7 биективны, так как  $\cos(\phi)$  монотонен на промежутке  $\phi \in [0, \pi]$ . Это свойство означает, что из матрицы  $G$  можно однозначно восстановить исходный временной ряд. Ещё одним важным свойством метода GAF является сохранение временной зависимости ряда: время увеличивается с перемещением позиции с левого верхнего угла матрицы  $G$  в правый нижний.

### 2.2.3 Moving Average Mapping

Методы кодирования могут строится на основе индикаторов технического анализа. Тайваньские исследователи в работе [5] предложили кодировать временной ряд с помощью скользящего среднего – одного из самых известных индикаторов. Такой метод кодирования получил название Moving Average Mapping.

Скользящая средняя имеет более плавную кривую, чем исходные данные, поэтому она имеет и большую временную зависимость. Для каждого значения ряда вычисляются скользящие средние с разными размерами фреймов. На основе этого строится матрица метода, которая аналогично матрице метода GAF будет преобразована в тепловую карту.

Рассмотрим временной ряд  $X = \{X_1, X_2, \dots, X_k\}$ . Матрица  $M$  для

этого временного ряда строится следующим образом:

$$M = \begin{pmatrix} MA_{1,1} & MA_{1,2} & \cdots & MA_{1,k} \\ MA_{2,1} & MA_{2,2} & \cdots & MA_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ MA_{k,1} & MA_{k,2} & \cdots & MA_{k,k} \end{pmatrix}$$

где  $MA_{i,j}$  — среднее между значениями с  $X_i$  по  $X_j$ , причем  $MA_{i,j} = MA_{j,i} \forall i, j \in [1, k]$ .

Ещё один кодировщик из этого семейства называется Double Moving Average Mapping. Вся разница заключается во входных данных: в случае с МАМ — это ряд значений закрытия торгов, а у ДМАМ — это ряд средних между значениями открытия и закрытия торгов. Матрица  $M$  для ДМАМ строится аналогично.

### Глава 3. Программная реализация и эксперименты

Рассмотрим ценовой ряд  $X = \{X_1, X_2, \dots, X_n\}$  (см. рис. 8). В данной работе горизонт прогнозирования  $r = 14$ , т. е. задача заключается в прогнозировании изменения фондового индекса через 14 дней. Количество классов  $s = 3$ : Up, Down и Flat. Параметры  $l_1 = -0.01$ ,  $l_2 = 0.01$ ,  $k = 14$ . Если  $\frac{X_{i+14} - X_i}{X_i} > 0.01$  ( $i \geq 14$ ), то отрезок  $\{X_{i-13}, X_{i-12}, \dots, X_i\}$  временного ряда  $X$  относится к классу Up, если  $\frac{X_{i+14} - X_i}{X_i} < -0.01$  — к классу Down, иначе — к классу Flat.

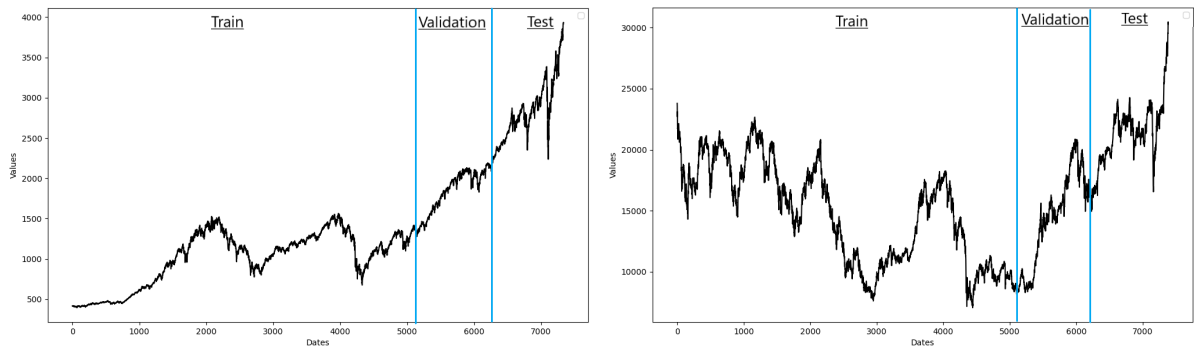
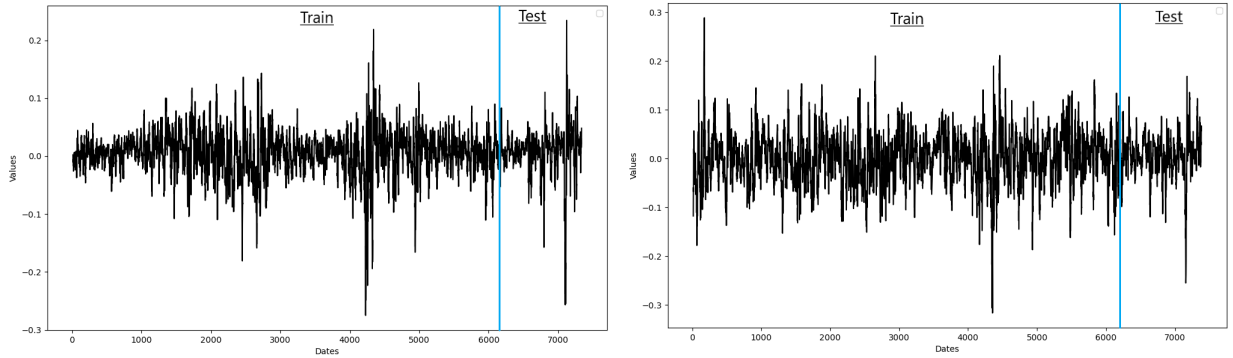


Рис. 8: Ценовые ряды для фондовых индексов S&P 500 (слева) и Nikkei (справа)

Для авторегрессионного подхода исходный ряд переводится в ряд относительных значений с лагом  $r = 14$  по формулам (1). Получим ряд  $Y = \{Y_1, Y_2, \dots, Y_{n-14}\}$ . Первые 14 значений ряда  $Y$  отбрасываются из рассмотрения. Если  $Y_i > 0.01$ , то отрезок  $\{X_{i-13}, X_{i-12}, \dots, X_i\}$  исходного временного ряда  $X$  относится к классу Up, если  $Y_i < -0.01$  — к классу Down, иначе — к классу Flat. Ряд  $Y$  был разделён на обучающее и тестовое множества в соотношении 0.85 : 0.15. На рис. 9 изображены графики рядов относительных значений для фондовых индексов S&P 500 и Nikkei.

Для нейросетевого подхода рассматривается исходный ряд в абсолютных значениях. При обучении свёрточной нейронной сети нельзя использовать тестовое множество, поэтому для контроля её обобщающей способности нужно взять часть обучающего множества, которая называется валидационным. Временной ряд был разделён на обучающее, валидационное и тестовое множества в соотношении 0.7 : 0.15 : 0.15. На рис. 8 изображены графики исходных ценовых рядов фондовых индексов S&P 500 и Nikkei.



**Рис. 9:** Ряды относительных значений для S&P 500 (слева) и Nikkei (справа)

Проверка качества моделей будет проводиться на тестовом множестве. Так как данная задача является задачей классификации, то для проверки качества моделей могут использоваться классические метрики оценки классификаторов.

### 3.1 Метрики

Наиболее наглядным показателем качества классификации является матрица неточностей, основная идея которой заключается в прямом подсчёте количества соотношений образцов класса  $C_i$  к классу  $C_j$ . В данной работе  $i, j \in \{1, 2, 3\}$ , т. е. имеем задачу классификации по 3 классам. Каждая строка в матрице неточностей — фактический класс, каждый столбец — спрогнозированный. Для удобства нормализуем матрицу неточностей по строкам.

Самая простая метрика для оценки классификатора — это правильность (*accuracy*). Правильность — доля правильных классификаций среди всех спрогнозированных объектов:

$$accuracy = \frac{1}{N} \sum_{i=1}^N I(C_i, P_i)$$

где  $I(C_i, P_i) = \begin{cases} 1, & \text{если } C_i = P_i \\ 0, & \text{если } C_i \neq P_i \end{cases}$ ;  $C_i$  — фактический класс  $i$ -ого объекта,  $P_i$  — спрогнозированный класс  $i$ -ого объекта,  $N$  — количество объектов.

Любую задачу классификации по  $s$  классам можно представить в виде  $s$  задач бинарной классификации, обозначив класс интереса как Positive, а объединение остальных как Negative. Введём общие понятия для бинарной классификации. True Positive (TP) — количество истинно положительных классификаций: соотнесение объектов класса Positive к классу Positive. True Negative (TN) — количество истинно отрицательных классификаций: соотнесение объектов класса Negative к классу Negative. False Positive (FP) — количество ложноположительных классификаций: соотнесение объектов класса Negative к классу Positive. False Negative (FN) — количество ложноотрицательных классификаций: соотнесение объектов класса Positive к классу Negative.

Для многоклассового случая эти величины можно подсчитать с помощью матрицы неточностей: True Positive — это значение на главной диагонали матрицы неточностей, False Positive — сумма остальных значений в столбце, False Negative — сумма остальных значений в строке, а True Negative — сумма всех оставшихся значений.

Теперь можно перейти к классическим метрикам бинарной классификации: точность (*precision*) и полнота (*recall*).

Точность (*precision*) — это доля истинно положительных классификаций среди всех объектов, спрогнозированных как положительные:

$$precision = \frac{TP}{TP + FP}$$

Полнота (*recall*) — это доля истинно положительных классификаций среди всех объектов положительного класса:

$$recall = \frac{TP}{TP + FN}$$

## 3.2 Особенности реализации

Эксперименты проводились на персональном компьютере с процессором Intel(R) Core(TM) i7-8750H 2.2GHz\*6, ОЗУ 8 Гб и видеокартой NVIDIA GeForce GTX 1050 Ti.

Все модели были реализованы на высокоуровневом языке программирования Python. Для реализации модели авторегрессионного скользящего среднего использовалась библиотека statsmodels [18]. Свёрточная нейронная сеть была реализована и обучена с помощью библиотек Keras [19] и Tensorflow [20]. Для реализации кодировщиков использовались библиотеки numpy [21] и pandas [22], для перевода матриц в тепловые карты — matplotlib [23], для метрик — модуль metrics из библиотеки scikit-learn [24].

### 3.3 Реализация авторегрессионного подхода

Рассмотрим ряды относительных значений  $Y$  (см. рис. 9). Проверим их на стационарность с помощью теста Дики—Фуллера. Для этого воспользуемся функцией adfuller из statsmodels. Статистика теста для ряда S&P 500 равна  $-12.9$ , для ряда Nikkei —  $-11.1$ . В обоих случаях статистики лежат левее критического значения при 5%-ом уровне значимости ( $-2.86$ ), следовательно, ряды стационарны, из чего следует, что порядок интегрирования  $d$  для них равен нулю.

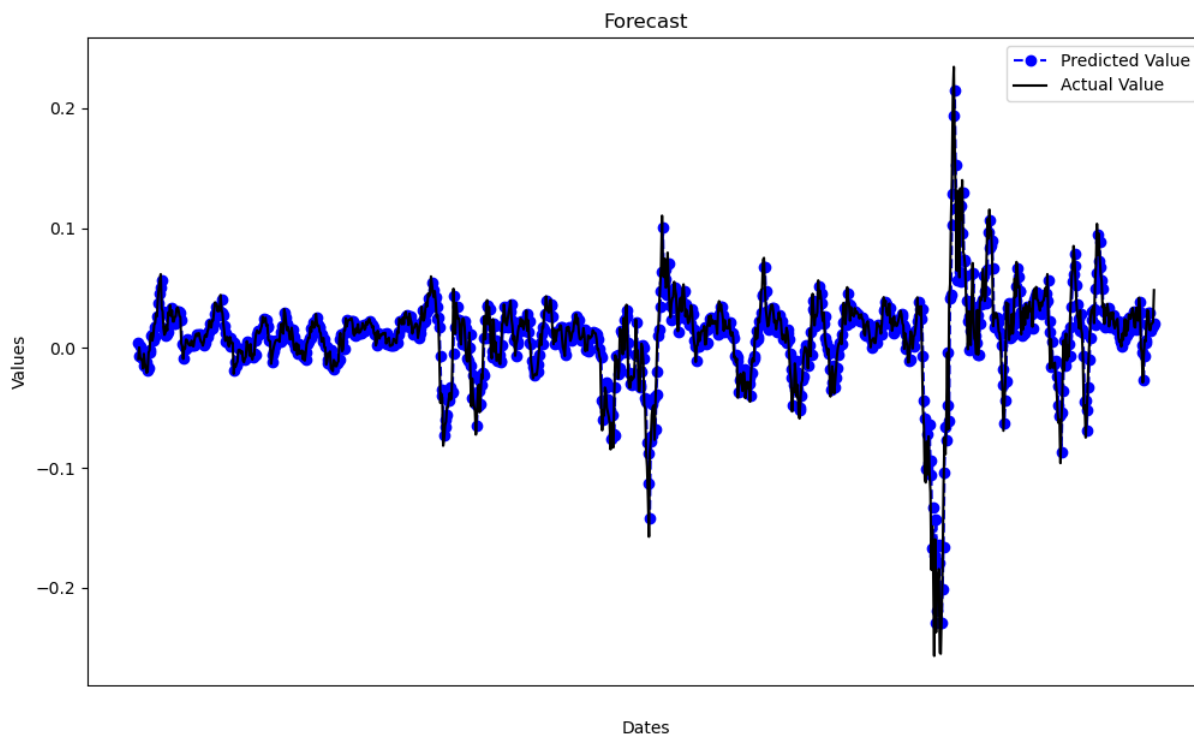
**Таблица 1:** Выбор оптимального порядка модели ARMA( $p, q$ )

| p, q   | AIC     |        | p, q   | AIC           |               |
|--------|---------|--------|--------|---------------|---------------|
|        | S&P 500 | Nikkei |        | S&P 500       | Nikkei        |
| (0, 0) | -27439  | -22096 | (2, 0) | -39913        | -35485        |
| (0, 1) | -33539  | -28361 | (2, 1) | -39911        | -35538        |
| (0, 2) | -36234  | -31333 | (2, 2) | -40133        | -35549        |
| (0, 3) | -37605  | -33000 | (2, 3) | -39981        | -35551        |
| (1, 0) | -39913  | -35487 | (3, 0) | -39912        | -35491        |
| (1, 1) | -39913  | -35485 | (3, 1) | -40134        | -35548        |
| (1, 2) | -39911  | -35490 | (3, 2) | -40434        | <b>-35953</b> |
| (1, 3) | -39929  | -35496 | (3, 3) | <b>-40440</b> | -35783        |

Далее, согласно методологии Бокса—Дженкинса, нужно определить порядок модели —  $p$  и  $q$ . Для этого нужно перебрать значения  $p$  и  $q$ , огра-

ничившись порядком 3 для каждого, и сравнить значения  $AIC$  для каждой модели-кандидата  $ARMA(p, q)$ . Проще всего сделать это с помощью функции `auto_arima` из библиотеки `pmdarima` [24]. Результаты поиска представлены в таблице 1. Для ряда S&P 500 наименьшее значение  $AIC$  у модели с параметрами  $p_0 = 3, q_0 = 3$ , для ряда Nikkei — с параметрами  $p_0 = 3, q_0 = 2$ .

Модель  $ARMA(p_0, q_0)$  строится на обучающем множестве с помощью функции `ARIMA()` из библиотеки `statsmodels`. Для проверки остатков построенной модели на соответствие белому шуму используется функция `acf(qstat=True)` из библиотеки `statsmodels`. Значения Q-статистики больше соответствующих р-значений, следовательно, автокорреляция отсутствует и ряд остатков является белым шумом.



**Рис. 10:** Прогнозирование фондового индекса S&P 500 на тестовом множестве моделью  $ARMA(3, 3)$

После построения модели прогнозируется следующее значение, которое будет соответствовать первому значению из тестового множества. Оно добавляется в обучающее множество, модель обучается на новых данных и прогнозируется следующее — второе — значение, и так до конца тестового множества. На выходе имеем набор спрогнозированных значений тестового

множества. На рис. 10 спрогнозированные значения сравниваются с фактическими. Можно заметить, что модель достаточно неплохо прогнозирует движение тренда, заметных выбросов нет.

С помощью спрогнозированных значений соответствующим отрезкам временного ряда сопоставляется один из классов Flat, Up или Down по правилу, описанному в начале главы. После этого можно строить метрики оценки классификаторов. Нормализованные матрицы неточностей для авторегрессионных моделей  $ARMA(p_0, q_0)$  приведены в таблице 2. Можно заметить, что оптимальные модели  $ARMA(p_0, q_0)$  для обоих фондовых индексов примерно одинаково прогнозируют каждый класс.

**Таблица 2:** Матрицы неточностей для моделей  $ARMA(p_0, q_0)$

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.69    | 0.24 | 0.08 | 0.69   | 0.16 | 0.15 |
| Up   | 0.11    | 0.89 | 0.01 | 0.13   | 0.86 | 0.02 |
| Down | 0.18    | 0.01 | 0.81 | 0.17   | 0.03 | 0.8  |

### 3.4 Реализация нейросетевого подхода

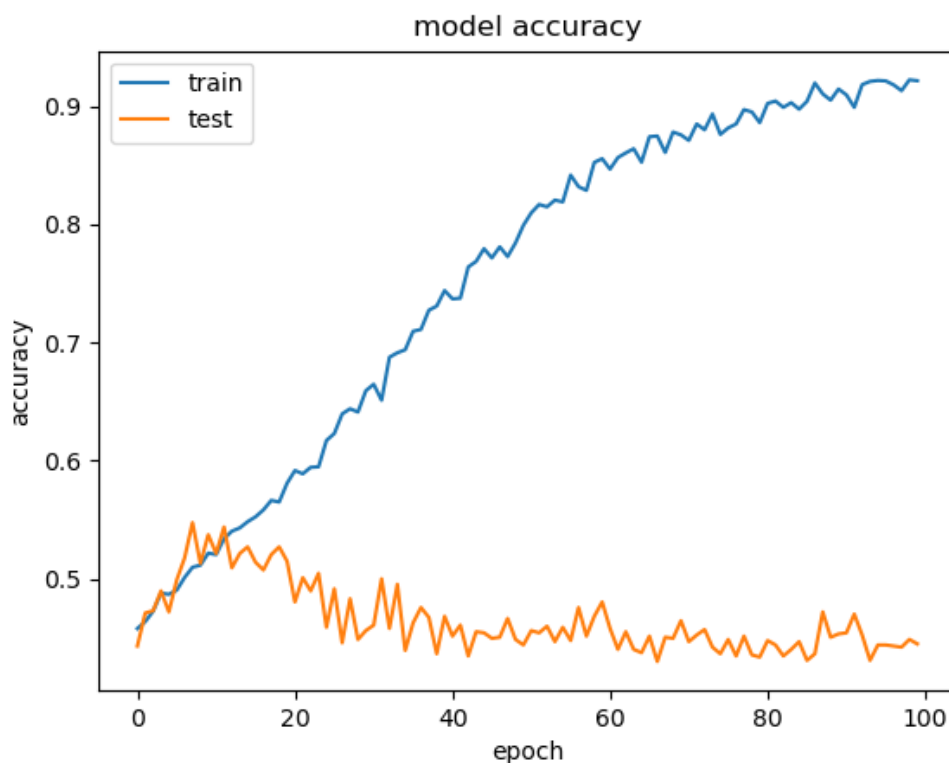
Обучающее, валидационное и тестовое множества были разбиты на окна размера  $k = 14$  и с лагом  $l = 1$ . Каждому окну сопоставляется метка принадлежности одному из классов Up, Down или Flat. После разметки данных размер каждого из множеств уменьшится на 14.

Размеры классов сильно отличаются друг от друга, поэтому их нужно сбалансировать. Для этого можно случайным образом отбросить часть объектов в больших классах.

В результате серии экспериментов в качестве свёрточной нейронной сети была выбрана VGG-подобная архитектура с 3 блоками из 1, 2 и 2 свёрточных слоёв размера  $3 \times 3$  с 32, 64 и 128 фильтрами соответственно.



Пулинговые слои использовали функцию максимума (max-pooling). Параметр dropout был установлен 0.25. На вход нейронной сети подаётся набор изображений обучающего и валидационного множеств и соответствующие им метки классов.

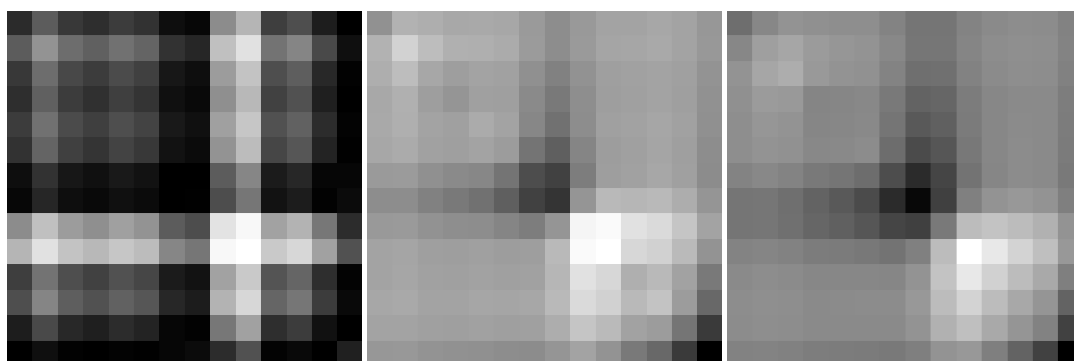


**Рис. 11:** Зависимость правильности от количества эпох для обучающего (синим) и валидационного (оранжевым) множеств для одноканальной модели с кодировщиком GAF на датасете S&P 500

Количество эпох обучения определяется по графикам зависимости правильности (*accuracy*) прогноза от количества эпох для валидационного и обучающего множеств. Момент, в котором начинается расхождение графиков и есть оптимальное количество эпох. Например, на рис. 11 приведены графики для одноканальной модели с кодировщиком GAF (на датасете S&P 500), на котором видно, что для этой модели данное число равно 15.

### 3.4.1 Эксперименты с одноканальными моделями

Каждое окно кодируется в изображение с помощью одного из методов, описанных в параграфе 2.2. Например, на рис. 12 приведены изображения закодированные методами Gramian Angular Field, Moving Average Mapping и Double Moving Average Mapping окна размера  $k = 14$ . Полученные изображения одноцветные (в градациях серого), следовательно, входные данные нейронной сети имеют 1 канал. Модели, которые используют один кодировщик, будем называть одноканальными.



**Рис. 12:** Изображения, полученные кодированием временного ряда размера  $k = 14$  методами GAF (слева), MAM (по центру), DMAM (справа)

Обучим свёрточную нейронную сеть на каждом из трёх наборов изображений, полученных с помощью методов GAF, MAM, DMAM (после балансировки). Далее с помощью построенных моделей сделаем прогноз на тестовом множестве. Нормализованные матрицы неточностей для каждой модели и каждого фондового индекса приведены в таблицах 3-5.

Одноканальная модель с кодировщиком GAF примерно одинаково прогнозирует классы для каждого из фондовых индексов, однако ошибается по-разному: если для S&P 500 класс Flat чаще путает с Up, то для Nikkei — с Down.

Одноканальная модель с кодировщиком MAM примерно одинаково прогнозирует классы Up и Down для каждого из фондовых индексов, однако класс Flat прогнозирует значительно лучше для S&P 500, чем для Nikkei. Матрица неточностей для S&P 500 почти достигает диагонального преобладания.

**Таблица 3:** Матрицы неточностей для одноканальных моделей с кодировщиком GAF

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.34    | 0.47 | 0.18 | 0.37   | 0.17 | 0.45 |
| Up   | 0.17    | 0.59 | 0.24 | 0.28   | 0.55 | 0.17 |
| Down | 0.14    | 0.23 | 0.63 | 0.16   | 0.08 | 0.77 |

**Таблица 4:** Матрицы неточностей для одноканальных моделей с кодировщиком МАМ

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.47    | 0.32 | 0.21 | 0.32   | 0.24 | 0.44 |
| Up   | 0.24    | 0.69 | 0.07 | 0.21   | 0.66 | 0.13 |
| Down | 0.2     | 0.12 | 0.68 | 0.18   | 0.09 | 0.73 |

**Таблица 5:** Матрицы неточностей для одноканальных моделей с кодировщиком ДМАМ

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.29    | 0.45 | 0.26 | 0.34   | 0.33 | 0.33 |
| Up   | 0.15    | 0.78 | 0.07 | 0.18   | 0.73 | 0.09 |
| Down | 0.16    | 0.16 | 0.67 | 0.16   | 0.14 | 0.7  |

Одноканальная модель с кодировщиком ДМАМ примерно одинаково прогнозирует классы для каждого из фондовых индексов. Данная модель

значительно лучше прогнозирует класс Up по сравнению с другими одно-канальными моделями.

### 3.4.2 Эксперименты с многоканальными моделями

Как уже было сказано в параграфе 2.1 входные данные первого слоя свёрточной нейронной сети могут иметь несколько каналов. По каждому каналу можно подавать изображения, полученные разными кодировщиками. Такие модели будем называть многоканальными.

Построим 3 двухканальные модели: с кодировщиками GAF и MAM, GAF и DMAM, MAM и DMAM, а также трёхканальную: со всеми тремя кодировщиками GAF, MAM и DMAM. Нормализованные матрицы неточностей для каждой многоканальной модели и каждого фондового индекса приведены в таблицах 6-9.

**Таблица 6:** Матрицы неточностей для двухканальных моделей с кодировщиками GAF и MAM

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.27    | 0.45 | 0.27 | 0.47   | 0.21 | 0.31 |
| Up   | 0.05    | 0.88 | 0.07 | 0.27   | 0.65 | 0.09 |
| Down | 0.13    | 0.16 | 0.71 | 0.28   | 0.09 | 0.63 |

Двухканальная модель с кодировщиками GAF и MAM хорошо прогнозирует классы Up и Down для обоих фондовых индексов, а класс Flat для Nikkei прогнозирует значительно лучше, чем для S&P 500. Матрица неточностей для Nikkei почти достигает диагонального преобладания.

Двухканальная модель с кодировщиками GAF и DMAM примерно одинаково прогнозирует классы для обоих фондовых индексов. Матрица неточностей для S&P 500 почти достигает диагонального преобладания.

**Таблица 7:** Матрицы неточностей для двухканальных моделей с кодировщиками GAF и DMAM

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.46    | 0.43 | 0.1  | 0.39   | 0.33 | 0.28 |
| Up   | 0.18    | 0.79 | 0.03 | 0.22   | 0.71 | 0.07 |
| Down | 0.31    | 0.16 | 0.52 | 0.23   | 0.14 | 0.63 |

**Таблица 8:** Матрицы неточностей для двухканальных моделей с кодировщиками MAM и DMAM

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.45    | 0.36 | 0.19 | 0.45   | 0.28 | 0.27 |
| Up   | 0.13    | 0.8  | 0.07 | 0.22   | 0.69 | 0.09 |
| Down | 0.22    | 0.14 | 0.64 | 0.27   | 0.12 | 0.61 |

**Таблица 9:** Матрицы неточностей для трёхканальных моделей с кодировщиками GAF, MAM, DMAM

| Факт | Прогноз |      |      |        |      |      |
|------|---------|------|------|--------|------|------|
|      | S&P 500 |      |      | Nikkei |      |      |
|      | Flat    | Up   | Down | Flat   | Up   | Down |
| Flat | 0.41    | 0.38 | 0.21 | 0.47   | 0.24 | 0.29 |
| Up   | 0.15    | 0.78 | 0.07 | 0.31   | 0.6  | 0.09 |
| Down | 0.16    | 0.16 | 0.67 | 0.28   | 0.09 | 0.63 |

Двухканальная модель с кодировщиками MAM и DMAM примерно одинаково прогнозирует классы для обоих фондовых индексов. Матрицы неточностей и для S&P 500, и для Nikkei почти достигают диагонального преобладания.

Трёхканальная модель с кодировщиками GAF, MAM и DMAM примерно одинаково прогнозирует классы Flat и Down для обоих фондовых индексов, а класс Up для S&P 500 прогнозирует значительно лучше. Матрица неточностей для Nikkei почти достигает диагонального преобладания.

### 3.5 Сравнение моделей

Все построенные модели сравниваются по метрикам бинарной классификации *precision* и *recall* на каждом классе, а также по метрике ассурасу, которая рассчитывается для всех трёх классов.

**Таблица 10:** Сравнение моделей по метрикам *precision* и *recall* на классе «Flat»

| Модель             | S&P 500   |        | Nikkei    |        |
|--------------------|-----------|--------|-----------|--------|
|                    | Precision | Recall | Precision | Recall |
| ARMA( $p_0, q_0$ ) | 0.625     | 0.685  | 0.634     | 0.687  |
| GAF                | 0.529     | 0.341  | 0.470     | 0.375  |
| MAM                | 0.524     | 0.469  | 0.464     | 0.324  |
| DMAM               | 0.488     | 0.294  | 0.508     | 0.338  |
| GAF + MAM          | 0.617     | 0.275  | 0.474     | 0.473  |
| GAF + DMAM         | 0.492     | 0.464  | 0.471     | 0.389  |
| MAM + DMAM         | 0.575     | 0.455  | 0.488     | 0.455  |
| GAF + MAM + DMAM   | 0.459     | 0.185  | 0.458     | 0.473  |

В таблице 10 сравниваются модели по предсказательной способности класса Flat с помощью метрик *precision* и *recall*. Как можно заметить, ARMA( $p_0, q_0$ ) прогнозирует значительно лучше всех остальных моделей для обоих фондовых индексов. Двухканальная модель с кодировщиками GAF и MAM на одном из датасетов имеет почти такую же точность, как

и  $ARMA(p_0, q_0)$ , однако полнота у неё значительно меньше. Это означает, что данная модель относит к классу Flat чаще всего именно объекты класса Flat, однако многие объекты этого класса классифицируются неверно. Это видно и по матрице неточностей (см. таблицу 6). Стоит также отметить двухканальную модель с кодировщиками MAM и DMAM, которая для обоих фондовых индексов имеет высокие показатели и по метрике *precision*, и по метрике *recall*.

**Таблица 11:** Сравнение моделей по метрикам *precision* и *recall* на классе «Up»

| Модель           | S&P 500   |        | Nikkei    |        |
|------------------|-----------|--------|-----------|--------|
|                  | Precision | Recall | Precision | Recall |
| $ARMA(p_0, q_0)$ | 0.899     | 0.886  | 0.893     | 0.858  |
| GAF              | 0.458     | 0.588  | 0.684     | 0.549  |
| MAM              | 0.611     | 0.692  | 0.667     | 0.664  |
| DMAM             | 0.565     | 0.782  | 0.607     | 0.731  |
| GAF + MAM        | 0.752     | 0.866  | 0.676     | 0.646  |
| GAF + DMAM       | 0.574     | 0.791  | 0.601     | 0.709  |
| MAM + DMAM       | 0.619     | 0.801  | 0.631     | 0.690  |
| GAF + MAM + DMAM | 0.418     | 0.654  | 0.641     | 0.601  |

В таблице 11 сравниваются модели по предсказательной способности класса Up с помощью метрик *precision* и *recall*. Здесь снова модель  $ARMA(p_0, q_0)$  показывает значительно лучшие результаты: около 89% всех объектов, отнесённых к классу Up, действительно относятся к этому классу, а около 86-89% объектов класса Up обнаружены верно. Среди нейросетевых моделей можно выделить двухканальную модель с кодировщиками GAF и MAM, которая тоже имеет достаточно высокие показатели по обоим метрикам. Кроме этого, одноканальная модель с кодировщиком DMAM и все двухканальные модели имеют высокие значения полноты при относительно невысоких точностях, т. е. эти модели часто верно обнаруживают класс Up, однако так же относят к нему и объекты других классов.

**Таблица 12:** Сравнение моделей по метрикам *precision* и *recall* на классе «Down»

| Модель             | S&P 500   |        | Nikkei    |        |
|--------------------|-----------|--------|-----------|--------|
|                    | Precision | Recall | Precision | Recall |
| ARMA( $p_0, q_0$ ) | 0.871     | 0.806  | 0.806     | 0.797  |
| GAF                | 0.583     | 0.627  | 0.541     | 0.766  |
| MAM                | 0.611     | 0.692  | 0.551     | 0.728  |
| DMAM               | 0.662     | 0.672  | 0.614     | 0.701  |
| GAF + MAM          | 0.662     | 0.711  | 0.602     | 0.632  |
| GAF + DMAM         | 0.789     | 0.522  | 0.628     | 0.628  |
| MAM + DMAM         | 0.699     | 0.637  | 0.624     | 0.609  |
| GAF + MAM + DMAM   | 0.562     | 0.582  | 0.613     | 0.632  |

**Таблица 13:** Сравнение моделей по метрике *accuracy*

| Модель             | Accuracy |        |
|--------------------|----------|--------|
|                    | S&P 500  | Nikkei |
| ARMA( $p_0, q_0$ ) | 0.824    | 0.798  |
| GAF                | 0.517    | 0.560  |
| MAM                | 0.612    | 0.568  |
| DMAM               | 0.581    | 0.587  |
| GAF + MAM          | 0.621    | 0.582  |
| GAF + DMAM         | 0.594    | 0.573  |
| MAM + DMAM         | 0.631    | 0.583  |
| GAF + MAM + DMAM   | 0.472    | 0.567  |

В таблице 12 сравниваются модели по предсказательной способности класса Down с помощью метрик *precision* и *recall*. Модель ARMA( $p_0, q_0$ ) и здесь показывает лучшие результаты. Среди нейросетевых моделей стоит отметить двухканальную модель с кодировщиками GAF и DMAM, которая имеет высокую точность на одном из датасетов. Другие двухканальные мо-



дели и одноканальная модель с кодировщиком DMAM также показывают достаточно высокие результаты по обеим метрикам.

В таблице 13 все модели сравниваются по метрике ассигасу. У модели ARMA( $p_0, q_0$ ) наибольшая доля правильных классификаций для обоих фондовых индексов. Среди нейросетевых моделей двухканальные модели чуть лучше одноканальных и трёхканальной.

## Выводы

В рамках данной работы были рассмотрены два подхода к прогнозированию изменения фондовых индексов: авторегрессионный и нейросетевой. Задача была представлена в виде задачи классификации по трём классам: Up, Down, Flat. Для проверки качества моделей использовались метрики оценки классификаторов: точность (*precision*), полнота (*recall*) и правильность (*accuracy*).

Авторегрессионный подход состоял в построении модели авторегрессионного скользящего среднего  $ARMA(p, q)$ . Нейросетевой подход включал в себя построение 7 моделей: трёх одноканальных, трёх двухканальных и одной трёхканальной. В качестве методов кодирования временных рядов в изображения использовались Gramian Angular Field, Moving Average Mapping и Double Moving Average Mapping. Эксперименты проводились для двух фондовых индексов — S&P 500 и Nikkei.

На каждом из двух датасетов авторегрессионный подход показал значительно более высокое качество прогнозирования всех классов, чем нейросетевой. Все построенные модели прогнозировали класс Flat хуже, чем классы Up и Down.

В зависимости от класса можно выбрать разные нейросетевые модели для прогнозирования. Если цель заключается в прогнозировании класса Up, то кроме авторегрессионной модели можно попробовать также двухканальную модель с кодировщиками GAF и MAM. А двухканальная модель с кодировщиками MAM и DMAM подойдёт лучше для прогнозирования классов Flat и Down.

Двухканальные модели в целом лучше одноканальных, но трёхканальная модель прогнозирует хуже остальных. Эксперименты с многоканальными моделями показали, что имеет смысл пытаться улучшить нейросетевой подход, комбинируя разные кодировщики с целью выявления оптимальной конфигурации.

Кроме того, выбор модели зависит от датасета. Например, лучшей по правильности (*accuracy*) среди нейросетевых моделей для фондового индекса S&P 500 является двухканальная модель с кодировщиками MAM

и DMAM, а для Nikkei – одноканальная модель с кодировщиком DMAM.

Повышать качество прогнозирования нейросетевым подходом можно различными способами. Например, реализацией новых методов кодирования временных рядов в серию изображений; применением известных кодировщиков не к исходному ценовому ряду, а к техническому индикатору этого ряда; выбором другой архитектуры нейронной сети; использованием ансамбля нейронных сетей и т. д.

## Заключение

В ходе исследования было построено 8 моделей прогнозирования изменения фондовых индексов с помощью двух подходов: авторегрессионного и нейросетевого. Задача прогнозирования была сформулирована в виде задачи классификации.

В качестве данных были выбраны 2 фондовых индекса: S&P 500 и Nikkei. Для каждого подхода была проведена предобработка и разметка данных.

Реализация авторегрессионного подхода включала в себя построение модели авторегрессионного скользящего среднего с предварительной проверкой временного ряда на стационарность и подбором оптимальных параметров, остатки модели проверялись на соответствие белому шуму. Спрогнозированные моделью значения были соотнесены одному из рассматриваемых классов.

Для реализации нейросетевого подхода временной ряд был переведён в серию изображений с использованием методов кодирования Gramian Angular Field, Moving Average Mapping и Double Moving Average Mapping. Была построена свёрточная нейронная сеть, на основе которой были реализованы 3 одноканальные модели, использующие в качестве входных данных изображения, полученные кодировщиками GAF, MAM или DMAM, а также 4 многоканальные модели, использующие комбинации этих кодировщиков.

Построенные модели сравнивались с помощью метрик оценки качества классификации *precision*, *recall* и *accuracy*. Авторегрессионный подход на тестовых данных показал себя лучше нейросетевого. Многоканальные нейросетевые модели оказались лучше одноканальных.

## Список литературы

- [1] Лозговцев Ю., Сафин В., Федоров А. Технический анализ: графический подход. // М.: Международная академия инвестиций и трейдинга. 2011-2012. 66 с.
- [2] Меладзе В. Э. Курс технического анализа. // М.: Серебряные нити. 1997. 272 с.
- [3] Whittle P. Hypothesis Testing in Time Series Analysis. // Almqvist & Wiksells boktr. 1951. Т. 4.
- [4] Wang Zh., Oates T. Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks. // AAAI Publications, Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015. P. 40-46.
- [5] Chen J., Chen W., Huang C., Huang S., Chen A. Financial Time-Series Data Analysis Using Deep Convolutional Neural Networks. // 7th International Conference on Cloud Computing and Big Data. 2016. P. 87-92.
- [6] Синсиэр М. Всё об индикаторах рынка. // пер. с англ. Антонова О., Калинина А. М.: Манн, Иванов и Фербер. 2013. 224 с.
- [7] Box G., Jenkins G., Reinsel G., Ljung G. Time Series Analysis: Forecasting and Control, 5th Edition. // John Wiley and Sons Inc., Hoboken, New Jersey. 2015. 712 с.
- [8] Dickey D., Fuller W. Distribution of the Estimates for Autoregressive Time Series with a Unit Root. // Journal of the American Statistical Association. 1979. P. 427-431.
- [9] Barra S., Carta S. M., Corrigan A., Podda A. S., Recupero D. R. Deep Learning and Time Series-to-Image Encoding for Financial Forecasting // IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 3. 2020. P. 683-692.

- [10] Жерон О. Прикладное машинное обучение с помощью Scikit-Learn и Tensorflow: концепции, инструменты и техники для создания интеллектуальных систем. // Пер. с англ. – СПб: ООО «Альфа-книга», 2018. 684 с.
- [11] LeCun Y., Boser B., Denker J., Henderson D., Howard R., Hubbard W., Jackel L. Backpropagation Applied to Handwritten Zip Code Recognition. // Neural Computation. No 1. 1989. P. 541–551.
- [12] Dumoulin V., Visin F. A Guide to Convolution Arithmetic for Deep Learning. // arXiv preprint arXiv:1603.07285. 2016.
- [13] Alapatt D., Mascagni P., Srivastav V., Padoy N. Artificial Intelligence in Surgery: Neural Networks and Deep Learning. // arXiv preprint arXiv:2009.13411. 2020.
- [14] Rumelhart D., Hinton G., Williams R. Learning Internal Representations by Error Propagation. // Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1. 1986. P. 318–362.
- [15] LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-based Learning Applied to Document Recognition. // Proceedings of the IEEE. 1998. P. 2278-2324
- [16] Nash W., Drummond T., Birbilis N. A Review of Deep Learning in the Study of Materials Degradation // npj Materials Degradation. 2. 10.1038/s41529-018-0058-x. 2018.
- [17] Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-scale Image Recognition. // arXiv preprint arXiv:1409.1556. 2014.
- [18] Statsmodels documentation. <https://www.statsmodels.org/v0.10.2/>
- [19] Keras documentation. <https://keras.io/api/>
- [20] Tensorflow documentation. [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs)
- [21] Numpy documentation. <https://numpy.org/doc/>

- [22] Pandas documentation. <https://pandas.pydata.org/docs/>
- [23] Matplotlib documentation. <https://matplotlib.org/stable/contents.html>
- [24] Scikit-learn: Machine Learning in Python documentation. [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)
- [25] Pmdarima documentation. <https://alkaline-ml.com/pmdarima/>