

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Применение нейронных сетей для прогнозирования финансовых временных рядов»

Студент

А.Д. Пищулев
(И.О. Фамилия)

(личная подпись)

Руководитель

М.А. Тренина

(ученая степень, звание, И.О. Фамилия)

Консультант

М.В. Дайнеко

(ученая степень, звание, И.О. Фамилия)

Аннотация

Тема выпускной квалификационной работы – «Применение нейронных сетей для прогнозирования финансовых временных рядов». Тема прогнозирования является актуальной, так как получение наиболее точного прогноза будущего позволяет принимать наиболее выгодные решения для достижения положительных результатов в любой сфере человеческой деятельности, в том числе, финансовой. В качестве объекта исследования выступает задача прогнозирования временных рядов. Предметом исследования является метод прогнозирования финансовых временных рядов с помощью использования нейронной сети.

Целью данной работы является разработка и программная реализация алгоритма предварительной обработки данных и последующего выполнения прогноза поведения финансовых временных рядов.

Для достижения поставленной цели необходимо выполнить ряд задач:

1. Исследовать предметную область;
2. Разработать алгоритм предварительной обработки данных;
3. Выбрать модель нейронной сети и подобрать ее параметры;
4. Выполнить программную реализацию описанных алгоритмов;
5. Выполнить тестирование разработанной программы;
6. Сформировать вывод о полученных результатах.

В первом разделе данной работы приводится теория, необходимая для понимания темы работы и используемых методов. Второй раздел посвящен описанию проектирования алгоритма прогнозирования и его программной реализации. В третьем разделе описывается тестирование разработанной программы и приведены результаты прогнозов и их оценки.

Пояснительная записка состоит из введения, 3-х разделов, заключения, списка использованной литературы, состоящего из 24 источников, и одного приложения. Текст работы представлен на 68 страницах, содержащих 38 иллюстраций и 50 формул.

Abstract

The title of the graduation work is Forecasting the financial time series by using the neural networks.

The author dwells on forecasting the time series by applying the neural network.

The graduation work consists of an introduction, 3 sections, a conclusion, 38 figures, 50 formulae, the list of 24 references and 1 appendix.

The object of the investigation is the task of forecasting the time series.

The subject of the graduation work is method of forecasting the financial time series by using the neural network.

The purpose of this research is to develop and implement a software algorithm for preprocessing and forecasting the time series.

In order to achieve the purpose, a number of objectives have to be attained:

1. to study the object of the research;
2. to develop a preprocessing algorithm: to select the methods for smoothing the time series and determine the necessary amount of information for qualitative forecasting;
3. to choose a neural network model and select its optimal parameters;
4. to perform the software implementation of the algorithms described;
5. to test the developed programme on different time series and compare the results;
6. to draw a conclusion about the results obtained.

The first section considers the theoretical foundations necessary to understand the topic of this graduation work and the methods used.

The second section is devoted to describing the design of the financial time series-forecasting algorithm and its software implementation.

The third section reveals testing of the developed programme on price time series of several companies and presents the results of their forecasting and evaluating.

Содержание

Введение.....	6
1 Исследование предметной области	8
1.1 Временные ряды.....	8
1.1.1 Понятие временного ряда	8
1.1.2 Классификация временных рядов	10
1.1.3 Модели временных рядов	12
1.2 Фильтрация временного ряда	15
1.3 Реконструкция аттрактора динамической системы	21
1.3.1 Понятие аттрактора	21
1.3.2 Метод Такенса для реконструкции аттрактора по временному ряду	22
1.3.3 Выбор оптимальной временной задержки лага τ	24
1.3.4 Выбор оптимальной размерности лагового пространства m 25	
1.4 Нейросетевое прогнозирование.....	27
1.4.1 Определение нейронной сети	27
1.4.2 Классификация искусственных нейронных сетей	29
1.4.3 Математическая модель искусственного нейрона	32
1.4.4 Обучение нейронной сети.....	37
1.4.5 Алгоритм Бройдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти L-BFGS	39
1.4.6 Оценка качества прогнозирования	42
2 Программная реализация алгоритма прогнозирования финансовых временных рядов	45

2.1	Описание программных средств	45
2.2	Фильтрация временного ряда	46
2.3	Реконструкция аттрактора динамической системы	49
2.3.1	Подбор оптимальной временной задержки лага	49
2.3.2	Подбор оптимальной размерности лагового пространства	51
2.4	Подбор параметров и обучение модели	58
2.5	Выполнение прогнозирования и оценка результатов	61
3	Тестирование программы и анализ результатов	64
3.1	Тестирование разработанной программы для прогнозирования временных рядов акций нескольких компаний	64
3.2	Анализ результатов прогнозирования	66
	Заключение	68
	Список используемой литературы	70

Введение

Задача прогнозирования поведения некоторой системы является актуальной в современном мире. Это может быть прогноз погоды, прогноз активности пользователей на сайте или прогноз изменения цен акций компании на бирже. Прогнозирование является одним из начальных этапов планирования различного направления (экономическое, техническое). Оценка возможного поведения системы позволяет принимать взвешенные решения, а также повысить эффективность выполняемой деятельности и снизить риски провала. Выполнение прогнозирования поведения финансовых рядов позволяет участникам фондовой биржи делать наилучшие решения по покупке или продаже акций. Выбор наиболее точного алгоритма и правильная предварительная обработка данных позволяет повышать качество получаемого прогноза.

Использование нейронных сетей для выполнения прогнозирования временных рядов является наиболее перспективным и точным в теории методом решения задачи прогнозирования поведения финансовых временных рядов, так как машинное обучение подразумевает самообучение программы. В данной работе будет использоваться нейронная сеть многослойного персептрона. Для повышения качества выполняемого прогноза над исходным временным рядом будут выполнены операции предварительной обработки: операция сглаживания временного ряда и анализ оптимального количества информации для обучения нейронной сети.

Актуальность бакалаврской работы обусловлена актуальностью задачи прогноза временных рядов в целом и необходимостью описания алгоритма для выполнения прогнозирования.

В качестве объекта исследования выступает задача прогнозирования временных рядов. Предметом исследования является метод прогнозирования финансовых временных рядов с помощью использования нейронной сети.

Целью данной работы является разработка и программная реализация алгоритма предварительной обработки данных и последующего выполнения прогноза поведения финансовых временных рядов. Программная реализация будет выполнена на языке программирования Python с использованием облачного сервиса Google Colab.

Для достижения поставленной цели необходимо выполнить ряд задач:

1. Исследовать предметную область;
2. Разработать алгоритм предварительной обработки данных: выбрать методы сглаживания временного ряда и определить необходимое количество информации для качественного прогнозирования;
3. Выбрать модель нейронной сети и подобрать ее оптимальные параметры;
4. Выполнить программную реализацию описанных алгоритмов;
5. Выполнить тестирование разработанной программы на разных временных рядах и сравнить результаты;
6. Сформировать вывод о полученных результатах.

1 Исследование предметной области

1.1 Временные ряды

1.1.1 Понятие временного ряда

Понятие временного ряда описывает некоторый набор информации, собранный за определенный промежуток времени о состоянии процесса, подлежащего исследованию. В финансовой системе в виде временного ряда записываются, например, изменения котировок акций компании за какой-то промежуток времени. На рисунке 1 представлен пример графика временного ряда. На данном изображении продемонстрирован график изменения индекса S&P 500 за один торговый день на фондовой бирже.

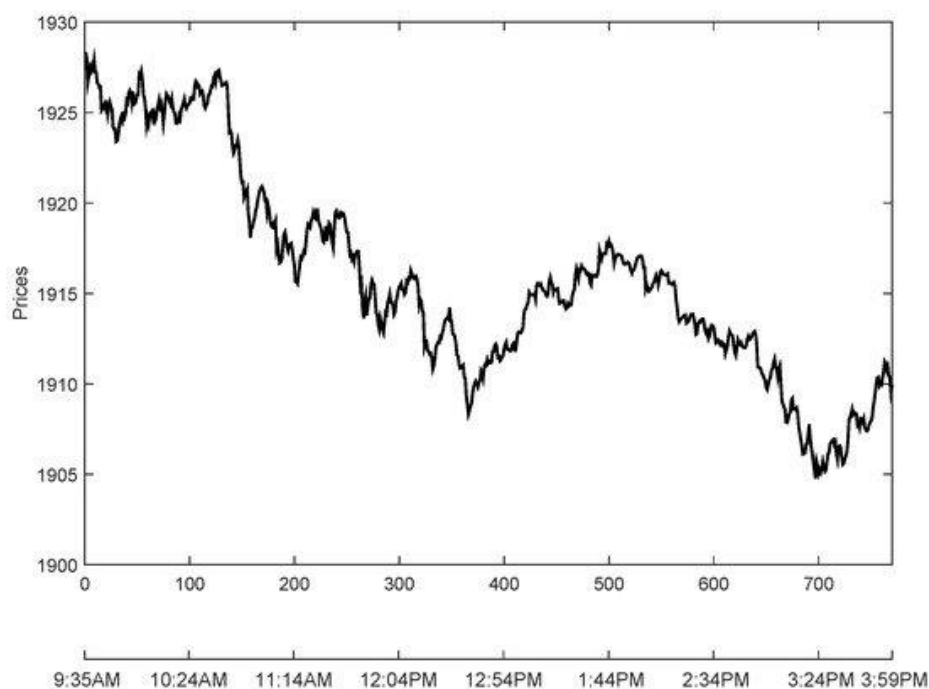


Рисунок 1 – Временной ряд изменений индекса S&P 500 за один день
7.08.2014 на фондовой бирже

В отличие от выборки, данные в которой никак не зависят друг от друга, временной ряд зависит от двух параметров: периода времени, за

который происходили изменения рассматриваемого процесса, и числовых показателей, характеризующих эти изменения. Таким образом, методы, используемые для анализа выборок данных, неприменимы для работы с временными рядами. Наблюдение за изменением показателя в определенный промежуток времени называется уровнем ряда.

Над временными рядами можно производить анализ при помощи определенных совокупностей методов математического и статистического анализа, при этом, данный анализ может быть направлен на выявление наличия возможной структуры временных рядов и, в следствии, их прогнозирование [9].

Анализ временных рядов состоит из выявления трендов ряда (определение закономерностей изменения наблюдаемого процесса), поиск сезонных компонент (повторяющихся циклов), поиск аномальных значений (уровней, оказывающих существенное влияние на значения ряда, но при этом не отвечающих возможностям исследуемой системы, т.е. ошибок технического или другого характера) и разложение ряда на составляющие, описанные выше: тренд, сезонность и ошибку [3].

Для прогнозирования временного ряда необходимо сформировать математическую модель исследуемого процесса. Прогнозирование временных рядов широко используется в инвестиционной финансовой деятельности, направленной на получение прибыли с вложенных денежных средств. Задачу прогнозирования временного ряда можно свести к поиску функции f_t (1):

$$y_{T+d} \approx f_T(y_T, \dots, y_1, d) \equiv \hat{y}_{T+d|T}, \quad (1)$$

где $d \in \{1, \dots, D\}$ – отсрочка прогноза,

D – горизонт прогноза.

Отсрочка прогноза определяет на сколько далеко будут предсказываться значения временного ряда. Для выполнения

прогнозирования временных рядов используются авторегрессионная модель (AR) и модель скользящее среднее (MA), а также их обобщения – модели ARMA и ARIMA [15].

1.1.2 Классификация временных рядов

Временные ряды классифицируются по наличию основной тенденции на стационарные и нестационарные. В стационарных рядах основной тренд отсутствует и математическое ожидание такого ряда равняется некоторой константе. На рисунке 2 продемонстрирован пример стационарного временного ряда.

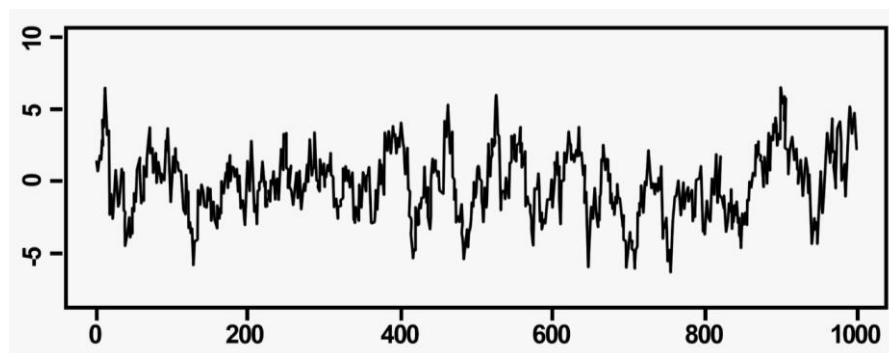


Рисунок 2 – Пример стационарного временного ряда

В нестационарных рядах можно выделить тенденцию к развитию ряда, например, возрастающий, убывающий ряд или некоторые циклические явления [18]. На рисунке 3 представлен пример нестационарного временного ряда с убывающей тенденцией.

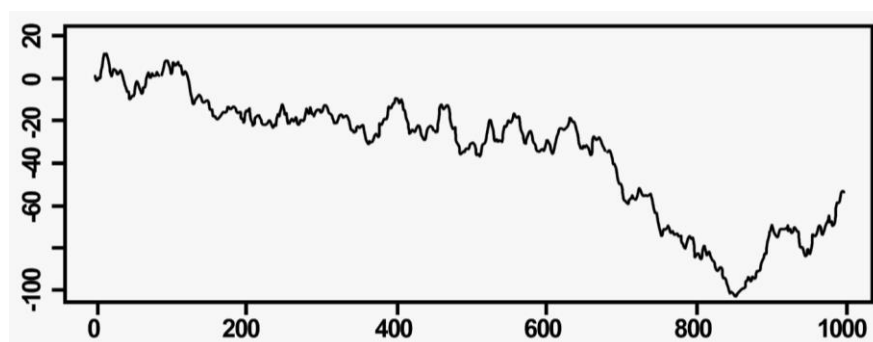


Рисунок 3 – Пример нестационарного временного ряда

По наличию сезонной компоненты временные ряды можно разделить на ряды с сезонностью и без сезонности. Сезонной компонентой называют некоторое периодическое явление, возникающее в данных в пределах некоторого периода, и являющееся серией циклических процессов, характеризующихся сезонными явлениями. Например, количество заказов и общая активность покупателей интернет-магазина будет резко увеличиваться в предпраздничные периоды и спадать после наступления праздников. Временной ряд активности покупателей интернет-магазина будет обладать сезонностью. На рисунке 4 представлен пример графика временного ряда с сезонностью.

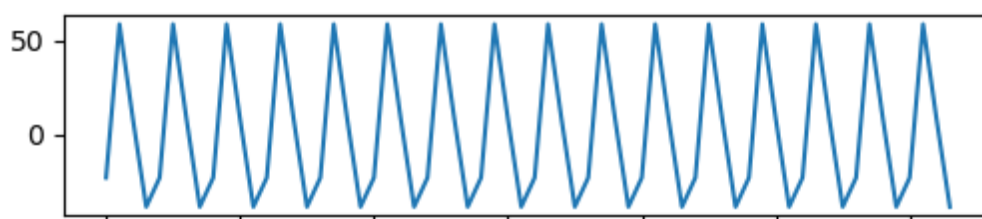


Рисунок 4 – Пример временного ряда с сезонностью

На рисунке 5 приведен пример графика временного ряда без сезонности. Изменения его значений никак не обусловлены в зависимости от какого-либо сезонного явления и являются случайными.

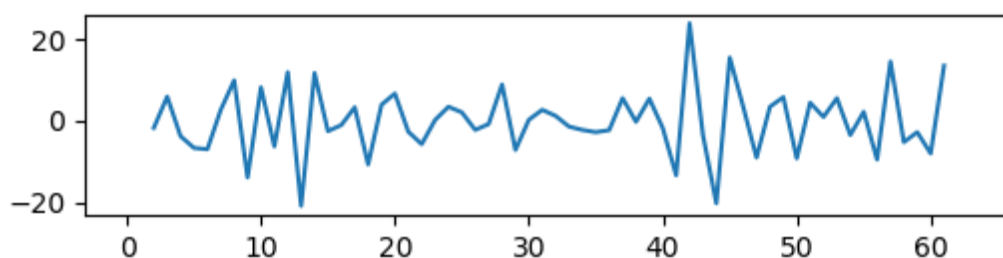


Рисунок 5 – Пример временного ряда без сезонности

Временные ряды могут обладать несколькими характеристиками одновременно, например, временной ряд может иметь сезонность и быть возрастающим.

1.1.3 Модели временных рядов

Существует две модели временных рядов с сезонностью. Внешний вид графика такого временного ряда зависит от выбранной модели. Значения уровней временных рядов зависят от показателей трендовой (T), сезонной (S) и случайной (E) компонент [5]. Если модель строится на основе суммы данных трех компонент, тогда это аддитивная модель (от англ. addition – сложение). Если компоненты умножаются, тогда это мультипликативная модель (от англ. multiply – умножение). Выбор модели составления временного ряда производится, основываясь на характере амплитуды сезонного хода. Амплитудой сезонного хода или амплитудой колебаний называют линию, соединяющую уровни временного ряда. Если амплитуда постоянна, т.е. имеет является постоянно возрастающей или постоянно убывающей, то выбирается аддитивная модель построения временного ряда. На рисунке 6 представлена аддитивная модель временного ряда с постоянно возрастающей амплитудой сезонного хода.

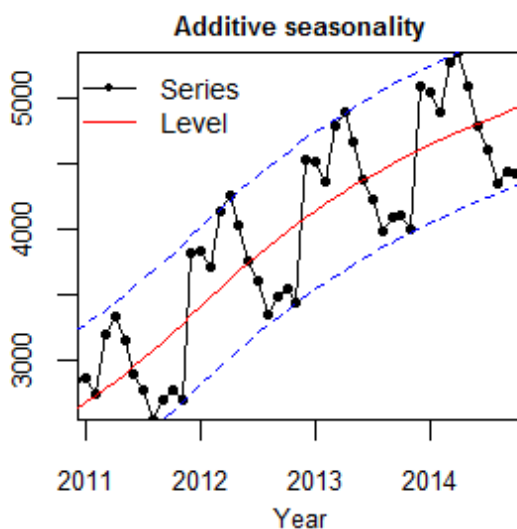


Рисунок 6 – Аддитивная модель временного ряда с сезонностью

Если амплитуда колебаний имеет свойство убывать и возрасти с течением времени, тогда выбирают мультипликативную модель построения временного ряда с сезонностью [8]. На рисунке 7 представлена аддитивная модель временного ряда с амплитудой колебаний, зависящей от уровня.

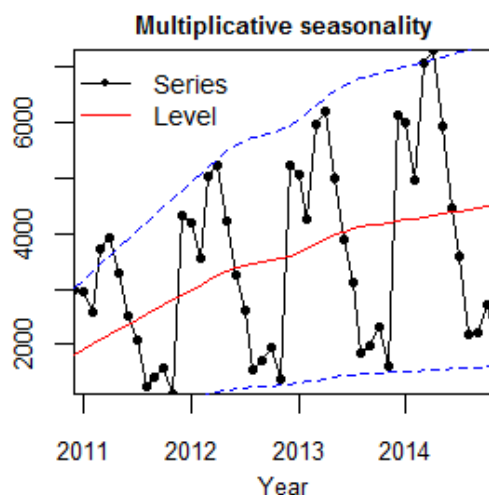


Рисунок 7 – Мультипликативная модель временного ряда с сезонностью

Для построения модели временного ряда используется следующий алгоритм:

1. Механическое выравнивание исходного временного ряда. Выравнивание или сглаживание временного ряда применяется для упрощения определения общей тенденции изменений значений временного ряда с помощью замены данных значений на теоретические значения, наименее подверженные случайным изменениям. Существует несколько методов сглаживания ряда: методов простой скользящей средней, метод экспоненциального сглаживания, метод медианного сглаживания. Обычно на практике для выполнения выравнивания используется метод простой скользящей средней, при котором исходные значения уровней ряда

заменяются на средние арифметические значения на заданных интервалах.

2. Вычисление значений сезонной компоненты (S). Сезонная компонента для аддитивной модели вычисляется как среднее значение всех значений ряда. Для этого используется показатель абсолютного отклонения $S \rightarrow S_{\Delta i}$. Сумма всех сезонных компонент должна быть равна нулю. Для мультипликативных моделей сезонная компонента вычисляется как урезанное среднее всех значений ряда. Для этого используется показатель индекса сезонности $S \rightarrow I_{si}$. Среднее значение всех сезонных компонент должна быть равна единице.
3. Устранение компоненты S из исходного ряда. Таким образом, получается временной ряд с выровненными значениями уровней (T плюс E для аддитивной модели и T умножить на E – для мультипликативной).
4. Выполнение аналитического выравнивания уровней трендовой (T) и случайной (E) компонент и получение значений трендовой компоненты T. Метод аналитического выравнивания основан на теореме Вейерштрасса, которая гласит следующее: «Любую функцию, непрерывную в некоторой окрестности точки t (на отрезке $[a, b]$) можно равномерно приблизить с любой точностью ε большим 0 многочленом $P_n(x) = a_0 + a_1x + \dots + a_nx^n$ заданной степени n , таким, что $\max_{a \leq x \leq b} |f(x) - P_n(x)| < \varepsilon$ ». Выбор многочлена осуществляется при помощи анализа сглаженного графика уровней временного ряда и определения вида трендовой модели: линейная, параболическая, гиперболическая, показательная.
5. Вычисление значений компонент T и E.
6. Вычисление абсолютных и относительных ошибок при помощи устранения из временного ряда сезонной (S) и трендовой (T) компонент.

На данном шаге необходимо также определить наличие автокорреляции. Для определения наличия автокорреляции используется критерий Дарбина-Уотсона [2]. Для начала предполагается, что величины остатков E_t для любого значения времени t не зависят от значений остатков для других t . Для проверки данной гипотезы используется следующая формула:

$$d = \frac{\sum_{t=2}^n (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^n \varepsilon_t^2}, \quad (2)$$

где $\varepsilon_t = y_t - \hat{y}_t$.

Если полученное значение критерия Дарбина-Уотсона равняется нулю, то говорят, что значения ошибок не содержат автокорреляции. Тогда исходные уровни ряда можно заменить данными значениями ошибок.

1.2 Фильтрация временного ряда

Исходный финансовый временной ряд является естественным сигналом, что, чаще всего, указывает на его нелинейность и нестационарность. Для выполнения прогнозирования поведения временного ряда редко используются начальные данные, так как они содержат шумовые компоненты хаотического характера, по которым невозможно построить корректной математической модели рассматриваемого явления. Для выполнения фильтрации временного ряда от шума и выявления тренда, по которому и будет выполняться прогнозирование, временной ряд рассматривают как сигнал. Фильтрация сигналов в частотной области выполняется с помощью преобразования Фурье, однако для работы с нестационарными и нелинейными данными используется эмпирический подход, при котором сигнал раскладывается на эмпирические функции и

функцию остатка. Для выполнения данного разложения используется итерационный алгоритм EMD (Empirical Mode Decomposition) [24].

Алгоритм модовой декомпозиции EMD основан на том, что любой случайный сигнал, имеющий, как минимум, 2 локальных экстремума, можно разделить на семейство внутренних эмпирических функции IMF и остаточную функцию тренда (2):

$$X(t) = \sum_{i=1}^n c_i(t) + r(t), \quad (3)$$

где $c_i(t)$ – внутренние эмпирические функции IMF,

$r(t)$ – остаточный тренд.

Норден Хуанг в 1998 году ввел понятие неявной эмпирической функции IMF, как колебательной волны, удовлетворяющей двум следующим требованиям:

1. Количество локальных экстремумов и нулей данной функции не должны отличаться более, чем на единицу;
2. В любой точке данной функции среднее значение огибающих локальных минимумов и локальных максимумов должно равняться нулю.

Огибающий сигнал – это волна, построенная по точкам локальных минимумов или локальных максимумов исследуемого сигнала. Для каждого сигнала, имеющего, как минимум, 2 локальных экстремума, можно построить нижний огибающий сигнал (построенный по точкам локальных минимумов) и верхний огибающий сигнал (построенный по точкам локальных максимумов). По нижнему и верхнему огибающим сигналам строится среднее значение огибающих, которое вычисляется как среднее значение данных сигналов. Разница между исходным сигналом и средним значением огибающих является потенциальной эмпирической модой IMF исследуемого сигнала.

Алгоритм EMD заключается в последовательном итерационном выделении эмпирических мод EMD исходного сигнала до тех пор, пока функция остатка не будет представлять из себя монотонную функцию. Данная операция выделения эмпирической моды EMD в исходном сигнале получила название sifting (просеивание). На первом этапе просеивания выполняется построение нижней и верхней огибающих сигнала. Для построения огибающих по точкам экстремума используется сплайн второго или третьего порядка. Вычисляется среднее значение огибающих m_1 и разность между сигналом и средним h_1 (3):

$$X(t) - m_1 = h_1 \quad (4)$$

Если полученный сигнал h_1 удовлетворяет условию IMF, процесс просеивания останавливается и сигнал h_1 считается найденной эмпирической модой IMF. Иначе для полученного сигнала h_1 также вычисляется среднее значение огибающих m_k и последовательно вычисляется сигнал h_k (4), пока для него не будет выполняться условие IMF.

$$h_{k-1} - m_k = h_k \quad (5)$$

Для проверки условия IMF будет вычисляться нормированная ошибка: отношение суммы квадратов средних огибающих на сумму значений полученного сигнала h_k (5), что позволит получать одинаковые значения суммы значений средних огибающих вне зависимости от сдвига ряда по оси OY.

$$e_k = \frac{m_k^2}{h_k^2} \quad (6)$$

Значение нормированной ошибки должно быть близко к нулю с заданной точностью. Алгоритм EMD продолжает выполнять просеивание до тех пор, пока из выходного сигнала возможно выделить очередную эмпирическую моду IMF, то есть, пока он имеет не менее 2 локальных экстремумов.

Для примера возьмем скорректированные цены закрытия акций компании Microsoft за период 01.01.2016 – 01.03.2016 с шагом в один день (рисунок 8). Скорректированная цена закрытия акции демонстрирует стоимость акции на момент закрытия торгового дня после выплаты дивидендов акционерам, что является фактической стоимостью акции в этот день.



Рисунок 8 – Скорректированная цена закрытия акций компании Microsoft за период 01.01.2016 – 01.03.2016

Локальными экстремумами называют точки изменения монотонности функции с убывающей на возрастающую (точка локального минимума) и с возрастающей на убывающую (точка локального максимума). На рисунке 9 на графике временного ряда скорректированных цен закрытия акций синим цветом отмечены точки локальных минимумов, красным цветом – точки локальных максимумов.

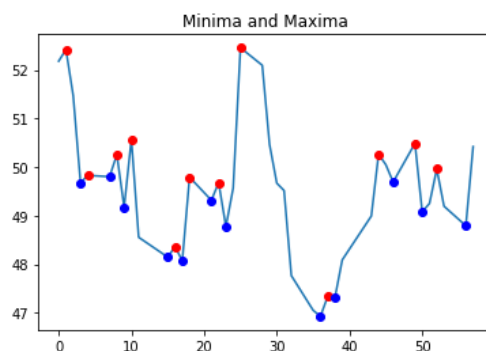


Рисунок 9 – Точки локальных минимумов и локальных максимумов

По точкам локальных экстремумов строятся нижний и верхний огибающие сигналы с помощью сплайна и среднее значений огибающих (рисунок 10):

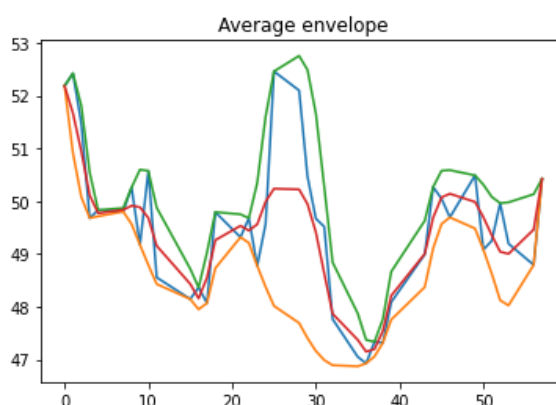


Рисунок 10 – Огибающие экстремумов и среднее значение огибающих

Полученное среднее значение огибающих является потенциальной эмпирической модой исходного сигнала. На рисунке 11 приведены графики полученных эмпирических мод IMF для исследуемого временного ряда скорректированных цен закрытия.

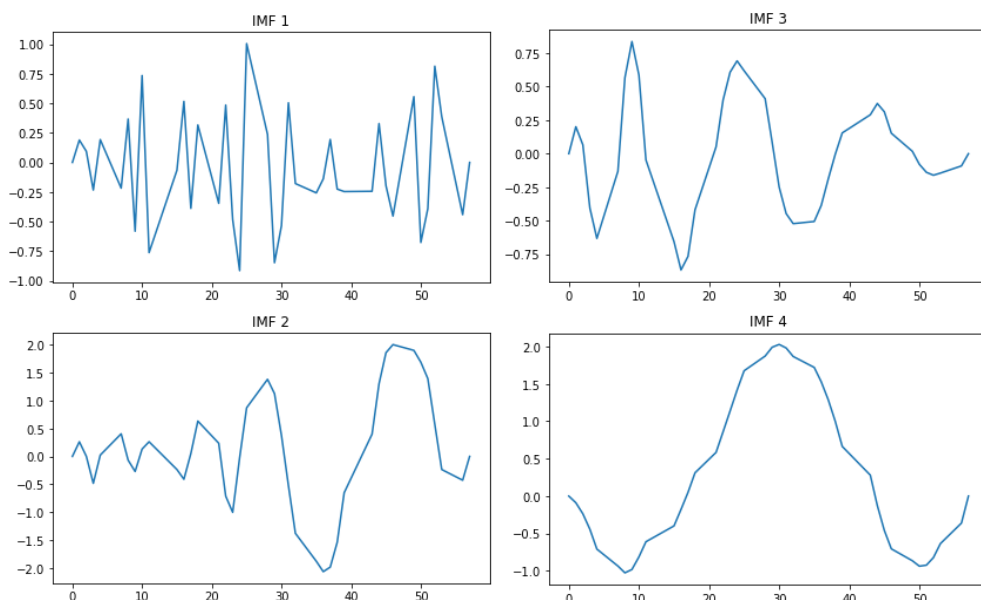


Рисунок 11 – Полученные эмпирические моды IMF

Обычно, одной итерации алгоритма просеивания недостаточно для того, чтобы полученный сигнал h_1 удовлетворял условию IMF, алгоритм выполняется, пока показатель нормированной ошибки (формула 6) не будет близким к нулю с заданной точностью. На рисунке 12 представлен график исходного временного ряда скорректированных цен закрытия и график полученного остаточного тренда r , который не содержит шумовых компонент.

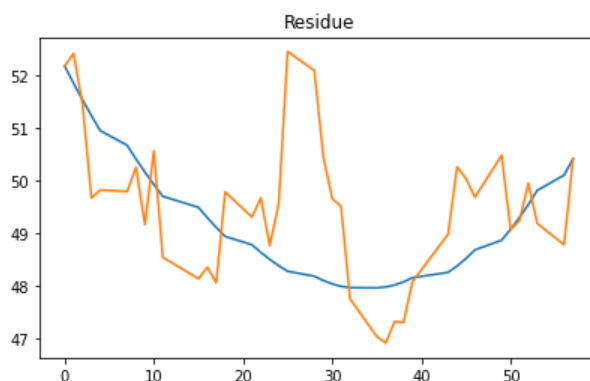


Рисунок 12 – График остаточного тренда временного ряда

Полученная функция тренда будет использоваться далее при нейросетевом прогнозировании поведения временного ряда.

1.3 Реконструкция аттрактора динамической системы

1.3.1 Понятие аттрактора

Аттрактором называется множество состояний, к которому, через определенный промежуток времени, стремится исследуемая динамическая система. Аттрактором может быть неподвижная точка или предельные замкнутые циклы, например, траектория движения маятника с трением о воздух является простейшим примером аттрактора [22]. Последовательность точек состояний системы в разные моменты времени в собственных координатах системы называется фазовой траекторией. Аттракторы с неустойчивыми траекториями движения внутри динамической системы хаотического характера называются странными аттракторами. Данный термин ввели ученые Рюэль и Такенс в 1971 году. Траектории движения внутри данного аттрактора устойчивы по некоторым локальным координатам и неустойчивы по другим и обладают высокой степенью чувствительности к незначительным изменениям входных данных, нарастающими во времени экспоненциально. Свойства странных аттракторов не изменяются при небольшом изменении динамической системы, поэтому странный аттрактор используется при определении динамического хаоса для моделирования поведения реальных детерминированных систем [4].

Одной из первых систем с странным аттрактором является система Лоренца, разработанную в 1963 году для решения проблемы конвекции воды в атмосфере. Данная модель получается из разложения скорости течения и температуры воды в ряды Фурье с усечением до первых и вторых гармоник и записывается в виде системы трех дифференциальных уравнений (6):

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases} \quad (7)$$

На рисунке 13 продемонстрировано графическое представление траекторий аттрактора Лоренца в двумерном пространстве.

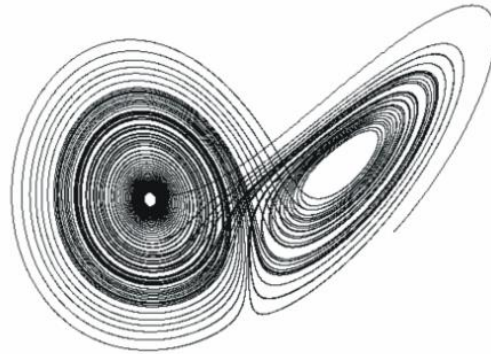


Рисунок 13 – Аттрактор Лоренца

Классическое значение параметров системы Лоренца являются σ равное 10, r равное 28, и b равное $\frac{8}{3}$.

1.3.2 Метод Такенса для реконструкции аттрактора по временному ряду

Аттрактор динамической системы можно использовать для прогнозирования состояний и поведения этой системы в будущем. По временному ряду наблюдений можно восстановить аттрактор динамической системы с помощью метода Такенса. Метод основан на теореме Такенса, предложенная Флорисом Такенсом в 1981 году и основанная на вложениях с запаздыванием или задержкой множеств, которая гласит, что, если временной ряд является произвольной функцией состояний системы, то существует такая глубина погружения, которая позволяет достичь однозначного предсказания следующего значения временного ряда [7].

Теорема Такенса гласит, что любое гладкое отображение n -мерного пространства X в m -мерное пространство Y будет задавать вложение множества A , являющегося компактным подмножеством пространства X , в пространство Y при условии, что

$$m > 2D_A + 1, \quad (8)$$

где D_A – фрактальная размерность множества A .

Существуют специальные методы, позволяющие восстановить параметры исследуемой динамической системы с помощью исследования траектории движения данной системы со следующими координатами:

$$Z_i^m = \{x_{\tau i}, x_{\tau i+1}, \dots, x_{\tau i+m-1}\}, \quad (9)$$

где τ – шаг задержки,

m – размерность пространства вложения.

Анализируемой динамической системой является поведение цен акций компании. Наблюдаемые значения записываются в виде одномерного временного ряда:

$$X = \{x(t) : 0 \leq t \leq n\}, \quad (10)$$

где n – количество наблюдений.

В данном случае, скорректированные цены закрытия акций компании. Необходимо выполнить реконструкцию аттрактора исследуемой динамической системы для дальнейшего прогнозирования ее поведения. Для этого нужно выполнить отображение координат $x(t)$ динамической системы в момент времени t в лаговое пространство с задержками:

$$X = [x(t), x(t + \tau), x(t + 2\tau), \dots, x(t + (m - 1)\tau)], \quad (11)$$

где $t \in [(m - 1)\tau + 1, n]$.

Лагом называется временной ряд, в котором значения взяты с некоторой заданной задержкой τ . Лаговое пространство – это матрица лаговых векторов, описывающая аттрактор с фрактальной размерностью d . Для погружения временного ряда скорректированных цен закрытия акций компании необходимо подобрать оптимальные значения параметров временной задержки лага τ и размерности лагового пространства (глубины погружения) m [16].

1.3.3 Выбор оптимальной временной задержки лага τ

Для подбора оптимального значения параметра временной задержки лага τ существуют несколько методов:

1. Подбор с помощью автокорреляционной функции АКФ, при котором в качестве временной задержки τ выбирается координата первого пересечения графика автокорреляции исходного сигнала с нулем;
2. Метод средней взаимной информации, при котором в качестве временной задержки τ выбирается первый локальный минимум функции взаимной информации $I(\tau)$.

Для определения оптимальной временной задержки τ будет использоваться метод средней взаимной информации [6]. Функция взаимной информации вычисляется с помощью следующей формулы:

$$I(\tau) = - \sum_{i,j} p_{ij}(\tau) \times \ln \frac{p_{ij}(\tau)}{p_i p_j}, \quad (12)$$

где $p_{ij}(\tau)$ – совместная вероятность точки p_i в фазовом пространстве p_j с задержкой τ .

Для временного ряда скорректированных цен закрытия акций компании Microsoft за период 01.01.2016 – 01.03.2016 график функции

средней взаимной информации от временной задержки τ представлен на рисунке 14.

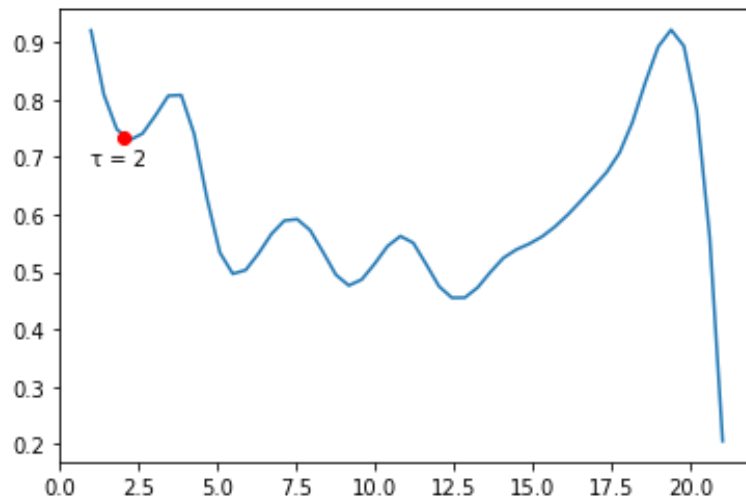


Рисунок 14 – График функции средней взаимной информации для временного ряда скорректированных цен закрытия акций

Своего первого локального минимума функция $I(\tau)$ достигает в точке τ равной двум, которая будет выбрана в качестве оптимального значения временной задержки для данного примера.

1.3.4 Выбор оптимальной размерности лагового пространства m

Для реконструкции аттрактора динамической системы по его траектории, записанной в виде временного ряда, необходимо также определить оптимальную размерность лагового пространства или глубину погружения m . Для определения значения этого параметра будет использоваться ложных ближайших соседей (false nearest neighbors – FNN) [23].

Алгоритм основан на подборе такого значения размера вложения m , при котором отношение числа ложных соседей $\bar{x}(i)$ для точки $\bar{x}(j)$ на общее число точек $\bar{x}(j)$ будет близким к нулю. Для этого для каждой точки $\bar{x}(j)$ вычисляется матрица расстояний до других точек в лаговом пространстве с размерностью m , изначально равной 1. Для каждой точки подбирается

ближайший сосед – точка с наименьшим значением расстояния до текущей [17]. Далее для каждой точки вычисляется отношение расстояния между точкой на следующем шаге $\bar{x}(j + 1)$ и ее ближайшим соседом $\bar{x}(i + 1)$ к расстоянию между точкой $\bar{x}(j)$ и ее соседом $\bar{x}(i)$ на текущем шаге:

$$R_j = \frac{\|\bar{x}(i + 1) - \bar{x}(j + 1)\|}{\|\bar{x}(i) - \bar{x}(j)\|} \quad (13)$$

Среди найденных R_j подсчитывается количество ложных соседей P_m , для которых R_j больше R , где R – некоторый заданный порог. В качестве значения порогового R будет использоваться число 10. Процент FNN для заданной глубины погружения m вычисляется как отношение числа ложных ближайших соседей к общему количеству точек (14):

$$F_m = \frac{P_m}{N} \quad (14)$$

В качестве метрики для вычисления расстояний между точками (лаговыми векторами) $\bar{x}(i)$ и $\bar{x}(j)$ используется матричная норма Фробениуса:

$$\|A\|_F = \sqrt{\sum_{i,j} |\bar{x}(i) - \bar{x}(j)|^2} \quad (15)$$

Алгоритм FNN повторяется для $m = m + 1$ до тех пор, пока не будет найдено такое m , при котором процент FNN F_m будет равным нулю.

После определения оптимальных значений временной задержки τ и размерности лагового пространства m возможно выполнить реконструкцию аттрактора динамической системы с помощью погружения временного ряда скорректированных цен закрытия акций компании в лаговое пространство.

Для этого формируется матрица с $m + 1$ столбцов $x(t), x(t - \tau), \dots, x(t - (m + 1)\tau), x(t + 1)$ и $(n - m\tau)$ строк. Данная матрица будет использоваться далее для обучения нейронной сети прогнозированию поведения цен акций компании.

1.4 Нейросетевое прогнозирование

Для прогнозирования финансового временного ряда был выбран метод прогнозирования с помощью построения регрессионной модели многослойного персептрона.

1.4.1 Определение нейронной сети

Искусственной нейронной сетью называется математическая модель и ее программная реализация для моделирования поведения реальных биологических нейронных сетей. Искусственная нейронная сеть состоит из соединенных между собой искусственных нейронов с помощью синаптических связей. Искусственный нейрон принимает на вход набор входных сигналов и вычисляет по ним с помощью внутреннего математического аппарата выходной сигнал. Выходные сигналы нейронов передаются по аксонам следующему слою нейронов нейронной сети. Синаптические связи нейронной сети имеют весовые коэффициенты, обозначаемые w , с помощью которых, подобно поведению биологической нейронной сети, возможно выполнять ослабление или усиление сигнала, передающегося по данным связям [10].

Характерной чертой искусственной нейронной сети является способность к самообучению при помощи корректировки весовых коэффициентов синаптических связей, что позволяет использовать нейронные сети для решения сложных задач различного характера, которые требуют аналитического размышления:

1. Задача классификации входных данных по группам из заранее известного набора групп;

2. Задача кластеризации – распределение входных данных на классы при неизвестном наборе классов и признаков, которыми должны обладать примеры, относящиеся к тому или иному классу;
3. Задача распознавания образов, например, текста, звука или объектов на изображении;
4. Аппроксимация непрерывной функции с заданной точностью;
5. Сжатие данных с помощью похожих данных или восстановление информации из зашумленных данных;
6. Прогнозирование поведения системы на основе нескольких предыдущих состояний данной системы, при наличии связи между данными показателями.

На рисунке 15 приведено графическое представление нейронной сети с одним скрытым слоем.

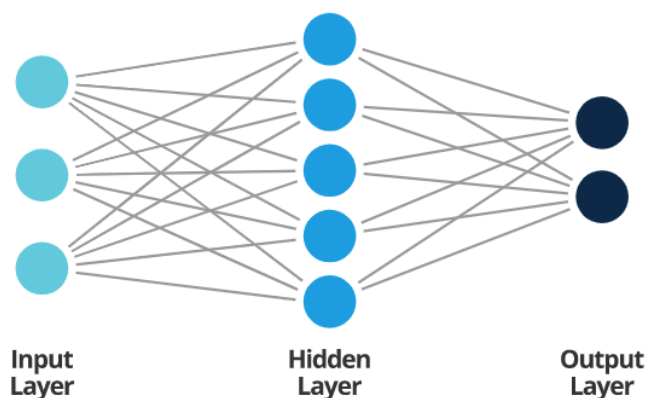


Рисунок 15 – Графическое представление нейронной сети с одним скрытым слоем

Нейронная сеть состоит из трех или более слоев искусственных нейронов: входной слой, обрабатывающий входные сигналы, скрытый слой или несколько скрытых слоев в зависимости от сложности модели, в данных слоях выполняются основные вычисления, и выходной слой, нейроны которого формируют результирующий сигнал. Использование нескольких скрытых слоев нейронной сети позволяет выявлять сложные зависимости в

входных данных. Задача выявления зависимостей некоторой системы сводится к задаче аппроксимации многомерной функции, которая описывает поведение этой системы. В задачах восстановления регрессионной модели, у которой есть несколько зависимых параметров и один независимый, выходная аппроксимирующая функция называется функцией регрессии [12].

1.4.2 Классификация искусственных нейронных сетей

Искусственные нейронные сети можно классифицировать по нескольким критериям. По способу обучения нейронные сети можно разбить на следующие архитектуры:

1. Нейронные сети, обучающиеся с учителем (наблюдаемое обучение), при этом для каждого элемента входных данных есть ожидаемое значение, которое должно получиться на выходе нейронной сети. Данная архитектура нейронных сетей используется при реализации персептрона Розенблатта и свёрточных нейронных сетей;
2. Нейронные сети, обучающиеся без учителя (самообучающиеся), при этом обучающая выборка не содержит ожидаемых выходных значений, с которыми сравниваются значения, получаемые нейронной сетью, и нейронная сеть должна выделить примеры со схожими характеристиками и сгруппировать их по классам. Данная архитектура нейронных сетей используется при реализации нейронных сетей адаптивного резонанса и самоорганизующийся карты Кохонена;
3. Нейронные сети, со смешанным обучением, когда часть весовых коэффициентов вычисляется с помощью наблюдаемого обучения, а другая часть корректируется без учителя. Данная архитектура нейронных сетей используется при реализации нейронных сетей радиально-базисных функций.

По способу настройки весовых коэффициентов нейронные сети разделяются на сети с фиксированными весами, заданными изначально, и

сети с динамическими весами, которые корректируются в процессе обучения сети.

По применяемой модели нейронные сети разделяются на:

1. Нейронные сети прямого распространения, когда синаптические связи между слоями нейронной сети направлены строго в одну сторону от входного слоя к выходному слою. На рисунке 16 представлен пример нейронной сети прямого распространения:

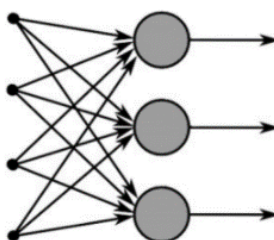


Рисунок 16 – Пример нейронной сети прямого распространения

2. Рекуррентные нейронные сети, у которых скрытые слои передают часть сигналов обратно нейронам входного слоя. На рисунке 17 продемонстрирован пример рекуррентной нейронной сети:

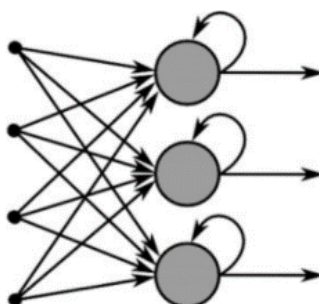


Рисунок 17 – Пример рекуррентной нейронной сети

3. Нейронные сети радиально-базисных функций, которые имеют один скрытый слой с нелинейной функцией активации, а весовые коэффициенты входного и выходного слоев равняются единице. На

рисунке 18 показан пример нейронной сети радиально-базисных функций:

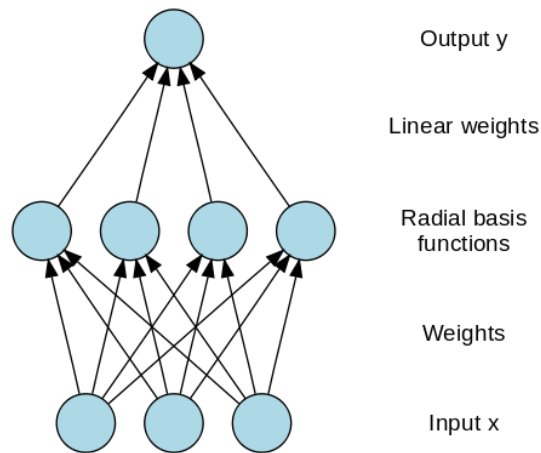


Рисунок 18 – Пример нейронной сети радиально-базисных функций

4. Самоорганизующаяся карта Кохонена (self-organizing map SOM), которая является самообучающейся нейронной сетью с одним входным и выходным слоем. Данная модель нейронной сети используется для решения задач кластеризации и распознавания образов. На рисунке 19 приведен пример самоорганизующейся карты Кохонена:

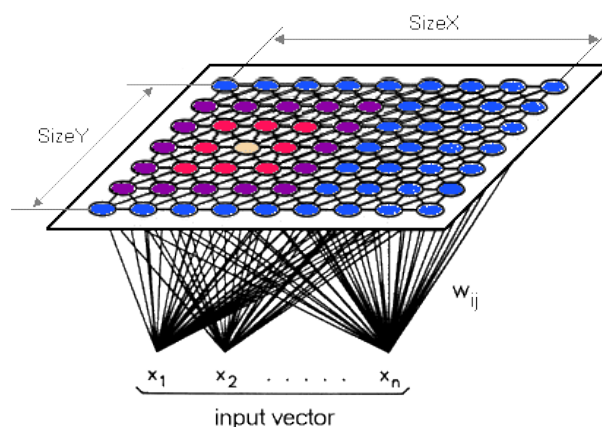


Рисунок 19 – Пример самоорганизующейся карты Кохонена

Для выполнения с помощью нейронной сети прогнозирования поведения временного ряда цен акций компании необходимо решить задачу регрессии, т.е. восстановление функции, описывающей поведение прогнозируемой системы. Для решения задачи регрессии в данной работе будет использоваться нейронная сеть прямого распространения многослойный персептрон. В качестве активационной функции для нейронов скрытых слоев будет использоваться сигмоидальная функция гиперболического тангенса [11].

1.4.3 Математическая модель искусственного нейрона

Искусственный нейрон является упрощенной версией естественного живого нейрона и состоит из нелинейной функции активации одного аргумента, на вход которой по синаптическим связям подаются сигналы, сумматора входных сигналов и единственного выходной сигнал, вычисляемый с помощью функции активации. Схематически искусственный нейрон изображен на рисунке 20:

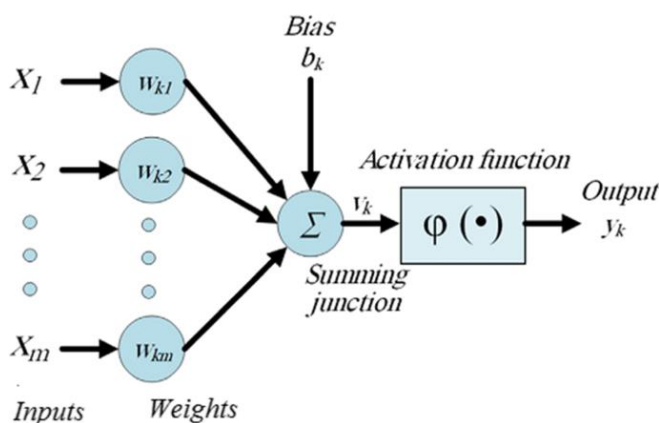


Рисунок 20 – Схематическое изображение искусственного нейрона

Синаптические связи нейрона имеют весовые коэффициенты w , задающиеся изначально или динамически подбирающиеся в процессе обучения модели [1]. При помощи весовых коэффициентов входные сигналы

нейрона усиливаются или ослабевают. Функция сумматора S выполняет сложение входных сигналов x_i , перемноженных на значения весовых коэффициентов w_i , в один сигнал для передачи его функции активации нейрона:

$$S = \sum_{i=1}^n w_i x_i + b, \quad (16)$$

где n – количество входных сигналов нейрона,

b – это коэффициент смещения.

Коэффициент смещения b (его также обозначают как w_0) выполняет роль константы линейной функции. Он используется для инициализации нейрона и позволяет выполнять сдвиг функции активации на постоянное значение. Это необходимо, например, когда на вход нейрона подаются только нулевые сигналы. Выходной сигнал нейрона y формируется с помощью функции активации $\varphi(S)$, в которую передается полученное сумматором значение (формула 17).

$$y = \varphi(S) \quad (17)$$

Функция активации $\varphi(S)$ отвечает за выявление зависимости выходного сигнала нейрона от его входных сигналов. В основном, эта функция является монотонной и возрастающей и имеет области выдаваемых значений от 0 до 1 включительно или от минус 1 до 1 включительно в зависимости от типа активационной функции [14]. Функции активации классифицируются по типу функции:

1. Ступенчатой или пороговой, при котором функция сохраняет постоянное значение до резкого перехода в некоторой точке T . На рисунке 21 представлен график ступенчатой функции активации.

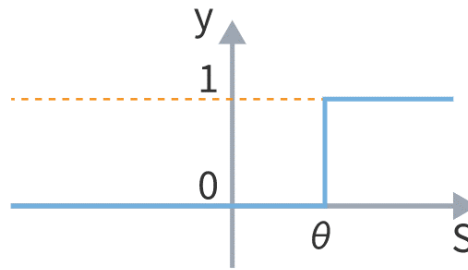


Рисунок 21 – График ступенчатой функции активации

Математическая формула ступенчатой функции активации имеет следующий вид:

$$\varphi(S) = \begin{cases} 1 & \text{if } S \geq T \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

2. Линейной, при которой выходной сигнал зависит от линейной функции $\varphi(S) = t \times S$, с заданным параметром t . Данный тип активационных функций в основном используется для нейронов входного слоя. На рисунке 22 представлен график линейной функции активации.

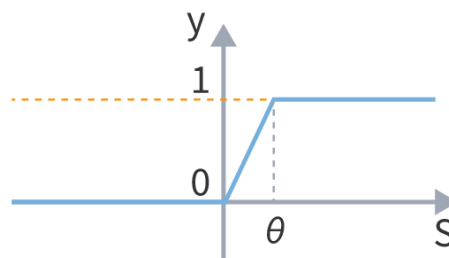


Рисунок 22 – График линейной функции активации

Математическая формула линейной функции активации имеет следующий вид:

$$\varphi(S) = \begin{cases} 0 & \text{if } S \leq 0 \\ 1 & \text{if } S \geq 1 \\ t \times S & \text{otherwise} \end{cases} \quad (19)$$

Сигмоидальной, при которой функция плавно изменяет значения от своего минимального до своего максимального значений, образуя S-подобную кривую сигмоиду. Данный тип активационных функций используется для нейронов внутренних скрытых слоев нейронной сети, так как производные таких функций являются квадратичными функциями, что значительно упрощает обучение сети алгоритмом обратного распространения ошибки, к тому же, такие функции являются устойчивыми к входным сигналам больших значений. Наиболее распространенными являются подтипы сигмоидальной функции: логистическая активационная функция и функция гиперболического тангенса.

Ниже представлена математическая формула логистической функции активации:

$$\varphi(S) = \frac{1}{1 + e^{-t \times S}} \quad (20)$$

С помощью изменения коэффициента t можно регулировать плавность перехода сигмоидальной функции. Формула активационной функции гиперболического тангенса представлена на формуле (21):

$$\varphi(S) = \frac{2}{1 + e^{-2t \times S}} - 1 \quad (21)$$

Гиперболический тангенс является скорректированной логистической функцией относительно оси абсцисс (область значений логистической функции от 0 до 1 включительно, область значений функции

гиперболического тангенса от минус 1 до 1 включительно). На рисунке 23 представлены графики логистической функции активации и функции гиперболического тангенса.

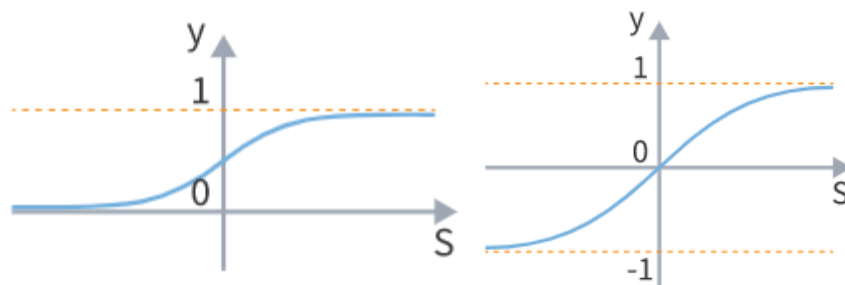


Рисунок 23 – Графики логистической функции активации и функции гиперболического тангенса

3. Радиально-базисной функцией, зависящей от величины расстояния между вектором входных сигналов S и заданным центром μ функции активации. На рисунке 24 представлен график радиально-базисной функции активации.

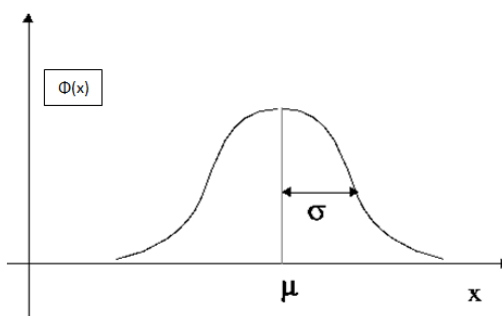


Рисунок 24 – График радиально-базисной функции активации

Математическая формула радиально-базисной функции активации имеет следующий вид:

$$\varphi(S) = e^{-\frac{(\|S-\mu\|-R)^2}{2\sigma^2}}, \quad (22)$$

где величина R характеризует сдвиг функции активации, а для вычисления расстояния $\|S - \mu\|$ используется метрика Евклида.

Как уже было описано выше, для нейронов внутренних скрытых слоев принято использовать сигмоидальную функцию активации, так как ее производная является квадратичной функцией, что облегчает процесс обучения такой нейронной сети, а также делает ее устойчивой к, так называемому, насыщению, когда сильные сигналы перекрывают более слабые сигналы при вычислении выходных сигналов нейронов. Для нейронной сети, используемой для прогнозирования поведения временных рядов цен акций компании, в качестве активационной функции для нейронов скрытых слоев будет использоваться сигмоидальная функция гиперболического тангенса.

1.4.4 Обучение нейронной сети

Целью обучения нейронной сети является такой подбор весовых коэффициентов для синаптических связей, при котором ошибка выдаваемых сетью результатов будет минимальной. Для этого, при выборе архитектуры обучения с учителем, входные данные делят на обучающие и тестовые (валидационные) в некотором соотношении, обычно, 4 к 1.

Для улучшения качества обучения модели исходные данные необходимо привести к стандартному диапазону, т.е. выполнить нормировку или стандартизацию данных. Если данные имеют нормальное распределение и диапазон значений достаточно большой, то необходимо выполнять процедуру стандартизации данных, в противном случае выполняется нормализация. Нормализация подразумевает приведение набора значений к диапазону от 0 до 1 включительно с помощью формулы (23):

$$y = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (23)$$

Стандартизация заключается в масштабировании наборов значений таким образом, чтобы среднее значение \bar{X} равнялось нулю, а величина стандартного отклонения σ_x равнялась единице. Стандартизированные значения вычисляются с помощью формулы (24):

$$y = \frac{x - \bar{X}}{\sigma_x} \quad (24)$$

Нейронная сеть подбирает весовые коэффициенты в соответствии с обучающими данными и проверяет результаты обучения на валидационных данных. Если ошибка прогноза сети на тестовых данных увеличивается, то произошло переобучение нейронной сети и обучающие данные просто запоминаются [13].

Самым простым алгоритмом обучения нейронной сети является алгоритм обратного распространения ошибки (backpropagation), основанный на стохастическом градиентном спуске, который позволяет корректировать весовые коэффициенты пропорционально их влиянию на ошибку получаемых сетью результатов. Для расчета показателя ошибки используется функция потерь $E(w)$ (loss function). Корректировка весовых коэффициентов осуществляется с помощью следующей формулы:

$$w_{k+1} = w_k - \alpha \nabla_w E(w), \quad (25)$$

где α – скорость обучения сети (числовой коэффициент).

Для вычисления новых весовых коэффициентов используется дифференциальный оператор Гамильтона ∇_w , обозначающий градиент функции потерь E . В качестве функции потерь E чаще всего используется функция среднеквадратической ошибки (mean squared error) (26):

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (26)$$

где y – результаты нейронной сети,

\hat{y} – ожидаемые результаты (обучающая выборка).

Также в качестве функции потерь используются функции кросс-энтропии, экспоненциальная функция, функция пророста информации (расстояние Кульбака-Лейблера).

Метод обратного градиентного спуска имеет ряд недостатков: медленная скорость обучения, проблема попадания в локальные минимумы целевой функции потерь, паралич сети, при котором весовые коэффициенты возрастают до больших значений, а значение производной функции активации уменьшается. При реализации реальных нейронных сетей используются более совершенные методы обучения.

Существует множество усовершенствованных методов обучения нейронной сети:

1. Методы стохастического градиентного спуска (метод сопряженных градиентов, метод градиентного спуска с учетом моментов, метод градиентного спуска с адаптивным обучением);
2. Импульсные методы (метод моментов);
3. Адаптивные методы (метод адаптивной оценки моментов, метод адаптивного градиента);
4. Квазиньютоновские методы (метод Бroyдена-Флетчера-Гольдфарба-Шанно).

Для обучения нейронной сети многослойный персептрон будет использоваться квазиньютоновский метод L-BFGS, который является модификацией метода Бroyдена-Флетчера-Гольдфарба-Шанно (BFGS) для решения задачи оптимизации весовых коэффициентов больших объемов [19].

1.4.5 Алгоритм Бroyдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти L-BFGS

Данный алгоритм оптимизации используется для нахождения локальных экстремумов нелинейного функционала. Для обучения нейронной сети необходимо выполнить оптимизацию следующего функционала (27):

$$\arg \min_w E(w) \quad (27)$$

Для нахождения минимума целевой функции потерь используется разложение в полином второй степени (28):

$$w_{k+1} = w_k + \nabla E^T(w_k)p + \frac{1}{2}p^T H(w_k)p, \quad (28)$$

где H – гессиан целевой функции потерь.

Гессианом называется определитель матрицы вторых частных производных целевой функции. При этом, используется не фактическое значение гессиана H , а его приближенное значение B_k . Обозначим обратный гессиан целевой функции потерь как $H_k = B_k^{-1}$.

Отличием алгоритма Бroyдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти от классического BFGS является наличие ограничения на количество используемых векторов шага s и разницы значений градиентов функции потерь u на каждой итерации. Для вычисления обратного гессиана H_k используется m последних векторов.

Описание алгоритма L-BFGS:

1. Инициализация счетчика $k = 0$;
2. Вычисление начального направления изменения весовых коэффициентов. Определим начальную точку p_k , в направлении которой будет выполняться поиск минимума целевой функции потерь (29):

$$p_k = \nabla E(w_k) \quad (29)$$

3. Уточнение направления изменения весовых коэффициентов (30):

$$\alpha_i = \frac{s_{k-i} * p_k}{s_{k-i} * y_{k-i}}, \quad p_k = p_k - \alpha_k y_{k-i}, \quad i \in [1, m], \quad (30)$$

$$p_k = \frac{s_{k-1}^T * y_{k-1}}{y_{k-1}^T * y_{k-1}}, \quad \text{для } k > 0,$$

$$\beta = \frac{y_{k-i}^T * p_k}{y_{k-i}^T * s_{k-i}}, \quad p_k = p_k + (\alpha_i - \beta) s_{k-i}, \quad i \in [m, 1]$$

4. Определение величины шага s_k на текущей итерации алгоритма и разницы значений градиента функции потерь y_k на текущей итерации выполняется по следующей формуле (31):

$$s_k = \eta p_k, \quad w_{k+1} = w_k + s_k, \quad y_k = \nabla E(w_{k+1}) - \nabla E(w_k) \quad (31)$$

Коэффициент η должен удовлетворять условиям Вольфе (32):

$$E(w_k + \eta p_k) \leq E(w_k) + c_1 * \eta * \nabla E^T(w_k) * p_k \quad (32)$$

$$\nabla E^T(w_k + \eta p_k) * p_k \geq c_2 * \nabla E^T(w_k) * p_k$$

Данный коэффициент подбирается с помощью линейного поиска при константах c_1 близкой к нулю и c_2 близкой к единице. Таким образом, подбирается такой коэффициент η , при котором значение целевой функции потерь в точке $w_k + \eta p_k$ минимально;

5. Инкремент счетчика $k = k + 1$ и переход на следующую итерацию алгоритма;
6. Шаги 2-5 выполняются последовательно в цикле, пока выполняется следующее условие (33):

$$\|\nabla E(w_k)\| > \varepsilon \quad (33)$$

7. Выходной вектор w_k является результатом процесса обучения нейронной сети методом Бройдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти.

Использование ограничения на m последних векторов s и u позволяет достичь понижения сложности алгоритма оптимизации. Сложность алгоритма становится линейной (34):

$$O(m \times k), \quad (34)$$

где k – это количество итераций алгоритма

Сложность классического метода Бройдена-Флетчера-Гольдфарба-Шанно является квадратичной (35):

$$O(k^2) \quad (35)$$

Алгоритм квадратичной сложности не подходит для оптимизации весовых коэффициентов больших объемов, поэтому в данной работе будет использоваться метод с ограниченным использованием памяти.

1.4.6 Оценка качества прогнозирования

После выполнения обучения выбранной модели нейронной сети выполнению задачи регрессии для предсказания поведения финансовых временных рядов акций компании необходимо разработать методику оценки качества этого прогнозирования. Для качественной оценки результатов обучения нейронной сети будет использоваться несколько показателей.

Для подбора оптимальных параметров модели нейронной сети будет использоваться среднеквадратическая ошибка (MSE):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (36)$$

где y – результаты нейронной сети,

\hat{y} – ожидаемые результаты из валидационной выборки.

Этот показатель позволяет выявить наличие больших ошибок в результатах выполняемого моделью прогноза, но не дает качественной оценки результатов прогноза в сравнении с тестовыми данными [20].

Для вычисления показателя точности прогноза нейронной сети будет использоваться нормированная среднеквадратическая ошибка или коэффициент детерминации R^2 (37):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (37)$$

Коэффициент детерминации R^2 изменяется в диапазоне чисел от 0 до 1 включительно. Чем ближе показатель коэффициента детерминации R^2 к единице, тем более точные результаты прогнозирования делает обученная модель нейронной сети [21]. Данный коэффициент будет использоваться для оценки качества выполняемого прогноза нейронной сетью.

1.5 Выводы по разделу

Таким образом, была исследована и описана теоретическая информация, необходимая для выполнения программной реализации алгоритма выполнения прогнозирования финансового временного ряда с помощью нейронной сети многослойного персептрона с обучением по методу Бройдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти L-BFGS и предшествующей предварительной обработкой входных данных. Предварительная обработка данных выполняется для повышения качества прогнозирования. Она включает в себя

фильтрацию исходного временного ряда с помощью метода эмпирической модовой декомпозиции EMD, погружение в лаговое пространство с предварительным подбором параметров временной задержки лага τ и размерности лагового пространства (глубины погружения) m , а также приведение данных к стандартному диапазону, т.е. выполнение нормировки или стандартизации обучающей выборки в зависимости от характера данных. Для выполнения оценивания результатов прогнозирования были выбраны формулы среднеквадратической ошибки для сравнения нескольких моделей и коэффициента детерминации для вычисления показателя точности прогноза.

В следующем разделе будет описана разработанная программа. Для выполнения программной реализации будет использоваться язык Python.

2 Программная реализация алгоритма прогнозирования финансовых временных рядов

2.1 Описание программных средств

Для выполнения программной реализации прогнозирования поведения финансовых временных рядов, реализующей описанные в теоретической части работы методы, был использован язык программирования Python и его основные библиотеки, список которых и краткое описание приведены далее.

Библиотека **scikit-learn** содержит множество моделей нейронных сетей для выполнения задач классификации, кластеризации и регрессии, а также множество функций для предварительной обработки начальных данных, таких, как нормализация и стандартизация данных. В качестве нейронной сети для выполнения прогнозирования в данной работе будет использоваться модель многослойного персептрона из данной библиотеки. Для работы с массивами и математическими матрицами используется библиотека **NumPy**, библиотека **SciPy** содержит набор алгоритмов и вспомогательных функций различных математических операций (например, нахождение экстремумов функции).

Библиотека **pandas** используется для наглядного вывода табличной информации, **matplotlib** содержит большой набор средств для визуализации данных в виде двумерных и трехмерных графиков. Данные библиотеки будут применяться в презентационной части программы.

Прогнозирование финансовых рядов будет выполняться на основе характеристик скорректированных цен закрытия акций компаний Microsoft, Apple, Intel, Disney, Visa и Coca-Cola за период 01.01.2016-01.01.2018 с шагом в один день. Информация о ценах акций компаний за выбранный промежуток времени была получена с помощью открытого источника Quandl. Процесс предварительной обработки исходного временного ряда, обучения

нейронной сети и получения результатов прогнозирования будет продемонстрирован пошагово для акций компании Microsoft.

2.2 Фильтрация временного ряда

В качестве временного ряда для выполнения прогноза будут использоваться данные скорректированных цен закрытия акций компании Microsoft за период 01.01.2016-01.01.2018 с шагом в один день. На рисунке 25 представлен график изменения скорректированных цен закрытия акций компании.

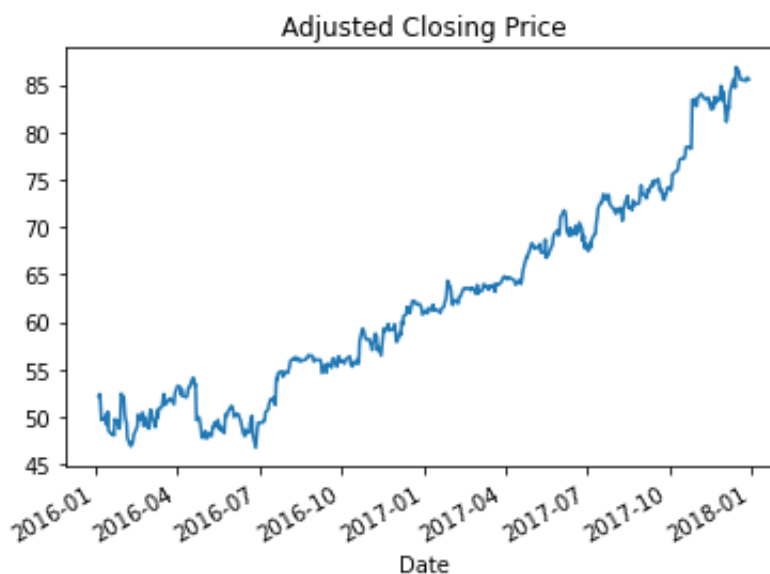


Рисунок 25 – Исходный график скорректированных цен закрытия акций компании Microsoft за период 01.01.2016-01.01.2018

Исходный график цен представляет собой зашумленные данные, по которому сложно производить прогнозирование. Необходимо выполнить фильтрацию исходного временного ряда и выделить линию тренда, на основе которого будет выполняться дальнейшее прогнозирование поведения временного ряда. Для выполнения фильтрации используется алгоритм разложения сигнала на эмпирические моды EMD, который был описан в

подразделе 1.2. Алгоритм модовой декомпозиции EMD основан на том, что любой случайный сигнал, имеющий, как минимум, 2 локальных экстремума, можно разделить на семейство внутренних эмпирических функции IMF и остаточную функцию тренда. На рисунке 26 представлена блок-схема алгоритма модовой декомпозиции EMD.

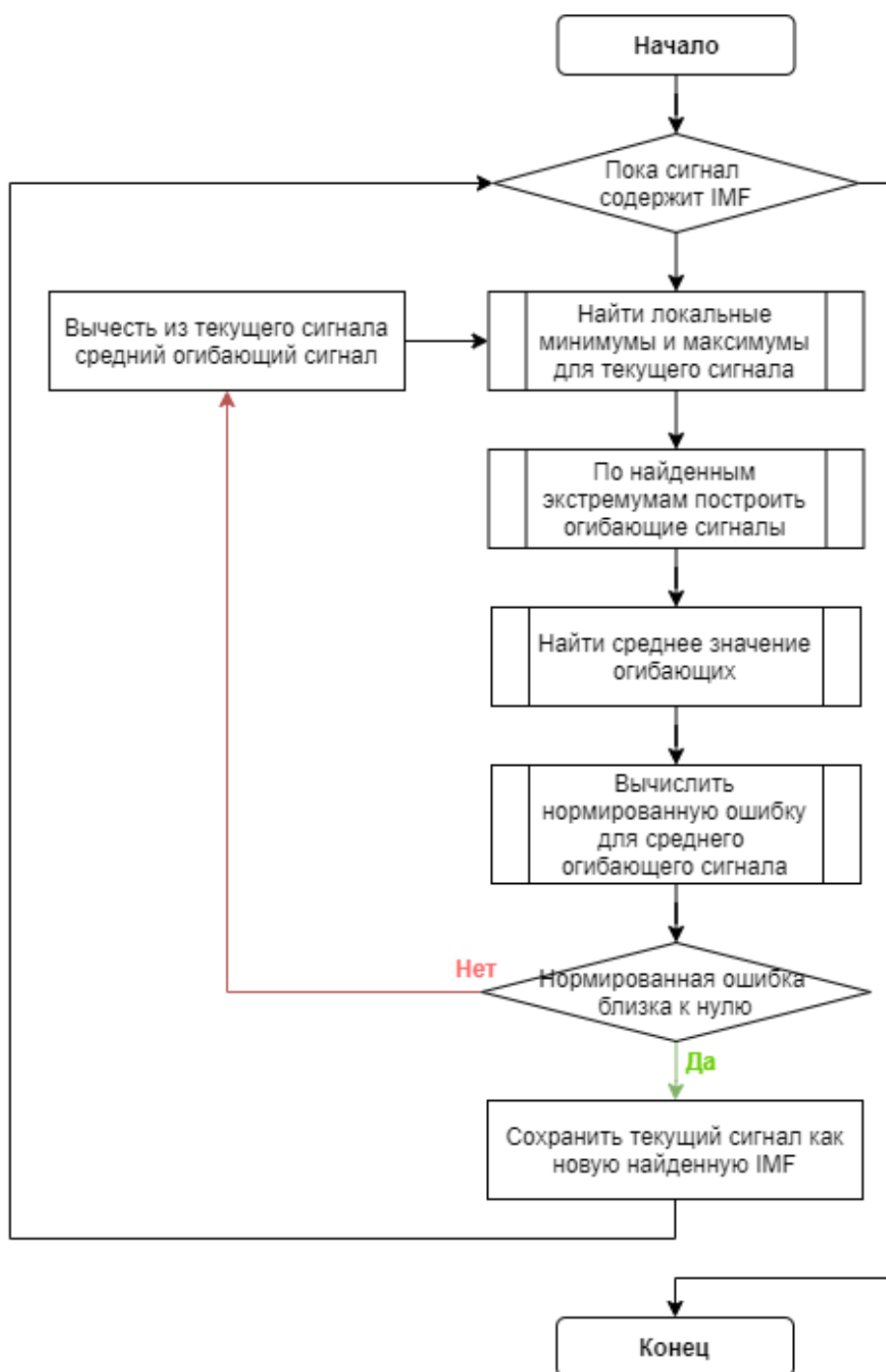


Рисунок 26 – Блок-схема алгоритма модовой декомпозиции EMD

На выходе алгоритма получается набор сигналов внутренних эмпирических мод IMF и сигнал остаточного тренда сигнала. Считается, что сигнал содержит в себе хотя бы одну внутреннюю моду IMF, если в нем есть, как минимум, 2 локальных экстремума. На рисунке 27 представлены графики результатов разложения временного ряда по методу EMD на 7 внутренних эмпирических мод IMF и функцию остаточного тренда.

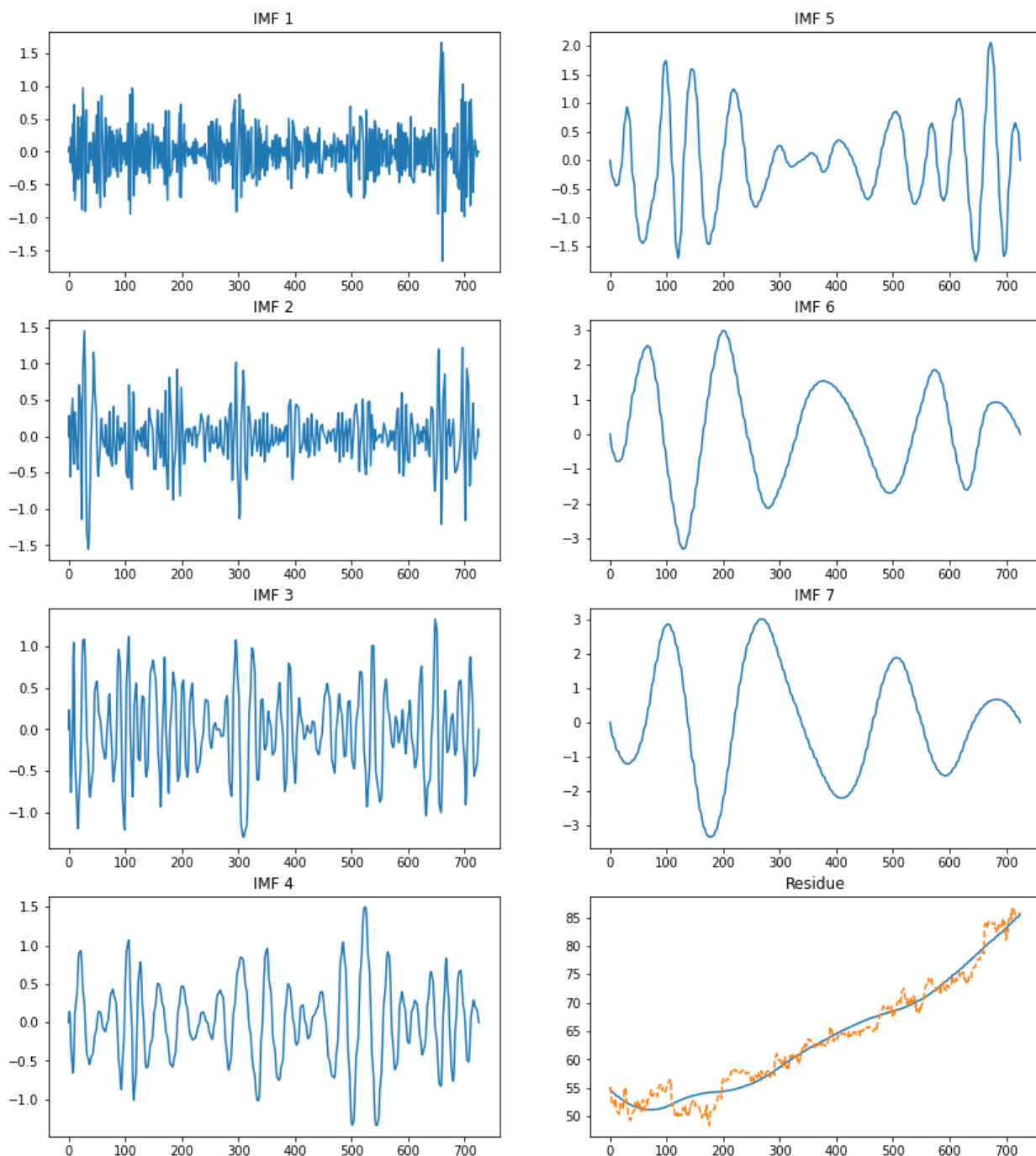


Рисунок 27 – Графики результатов разложения исходного временного ряда

На последнем графике в правой колонке синим цветом отмечена линия остаточного тренда, а оранжевым – исходный сигнал. Для прогнозирования будет использоваться полученная линия тренда, которая является результатом фильтрации временного ряда по методу EMD.

2.3 Реконструкция аттрактора динамической системы

2.3.1 Подбор оптимальной временной задержки лага

Как описано в подразделе 1.3, для успешного прогнозирования по временному ряду наблюдений можно восстановить аттрактор динамической системы с помощью метода Такенса. Для этого нужно выполнить отображение координат $x(t)$ динамической системы в момент времени t в лаговое пространство, для чего необходимо подобрать оптимальные значения параметров временной задержки лага τ и размерности лагового пространства (глубины погружения) m .

Для определения оптимальной величины задержки лага τ используется метод средней взаимной информации. Перед выполнением поиска оптимального значения задержки над сигналом необходимо предварительно произвести операцию нормировки. Алгоритм метода средней взаимной информации заключается в вычислении значений функции взаимной информации для значений показателя τ в некотором диапазоне. Формула функции взаимной информации приведена в пункте 1.3.3. Для вычисления значения функции взаимной информации исходный сигнал x разбивается на m отрезков. Количество отрезков вычисляется по следующей формуле:

$$m = \left\lceil \frac{\max x - \min x}{2 * IQR(x) * n^{-\frac{1}{3}}} \right\rceil, \quad (38)$$

где n – размер сигнала x ,

$IQR(x)$ – функция интерквартильного размаха.

Для каждого отрезка вычисляется значение вероятности попадания элемента на отрезок $p(A)$. Также рассчитывается матрица совместных вероятностей при заданной задержке τ . Совместная вероятность $p(AB)$ вычисляется как отношение количества элементов, принадлежащих отрезку A и B одновременно, к разнице общего количества элементов n и заданной лаговой задержки τ . На рисунке 28 представлена блок-схема алгоритма поиска оптимального значения задержки τ по методу средней взаимной информации.

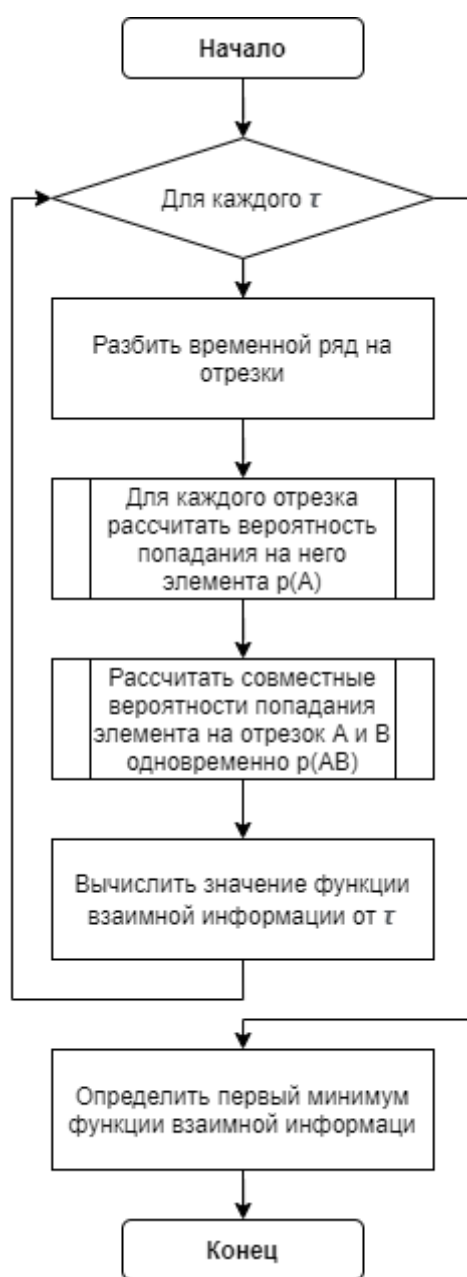


Рисунок 28 – Блок-схема метода средней взаимной информации

Для исходного временного ряда скорректированных цен закрытия акций компании Microsoft график функции средней взаимной информации $I(\tau)$ от временной задержки τ представлен на рисунке 29.

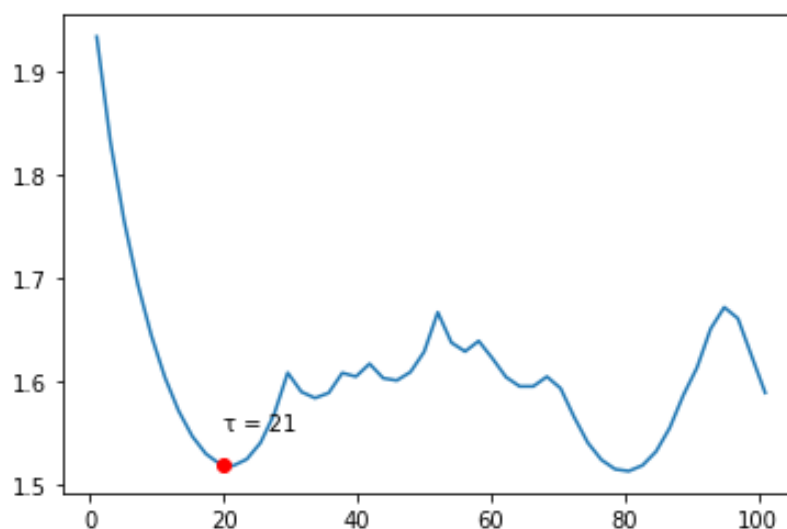


Рисунок 29 – График функции средней взаимной информации $I(\tau)$ для исходного временного ряда

Своего первого локального минимума функция $I(\tau)$ достигает в точке τ равной 21, которая будет выбрана в качестве оптимального значения временной задержки.

2.3.2 Подбор оптимальной размерности лагового пространства

Для подбора оптимальной глубины погружения m используется метод ложных ближайших соседей FNN. Алгоритм выполняет последовательное увеличение счетчика m величины глубины погружения до тех пор, пока процент FNN для текущей глубины не будет равняться нулю.

На рисунке 30 представлена блок-схема алгоритма подбора оптимального значения глубины погружения лагового пространства m с помощью метода ложных ближайших соседей для реконструкции аттрактора

динамической системы по временному ряду, описывающему траекторию движения данной системы.

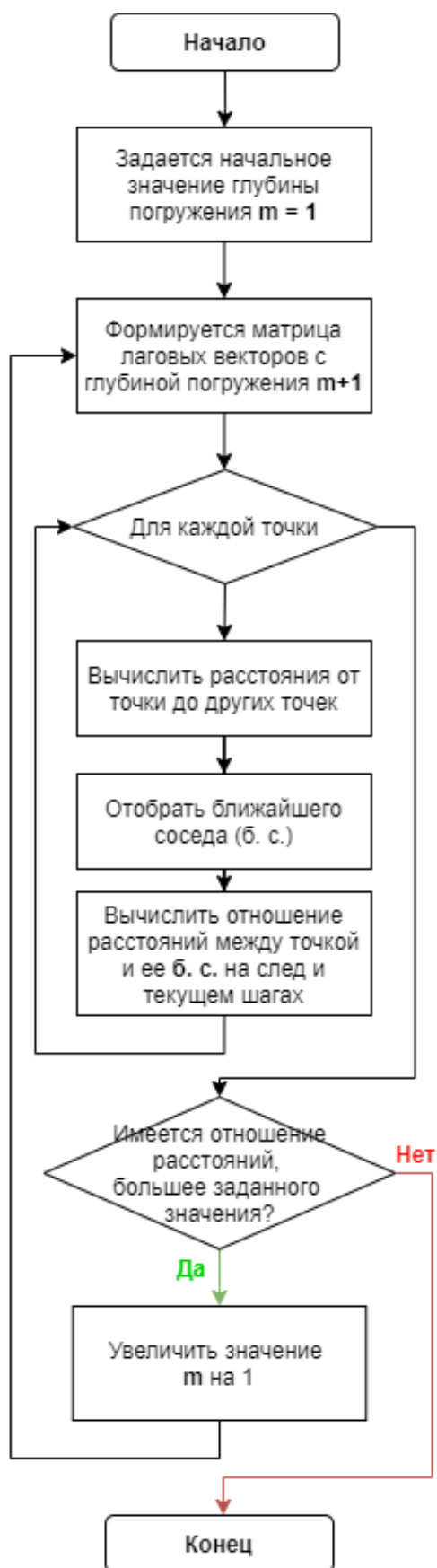


Рисунок 30 – Блок-схема алгоритма ложных ближайших соседей FNN

Для примера вычислим оптимальную глубину погружения m для временного ряда (39) при заданной временной задержке τ равной 3.

$$[77.17 \quad 61.74 \quad 51.54 \quad 73.34 \quad 62.43 \quad 69.04 \quad 53.6 \quad 51.2 \quad 51.47 \quad 51.58] \quad (39)$$

Начальным значением m является 1, поэтому расширенная матрица лаговых векторов $m + 1$ будет иметь следующий вид:

$$\begin{bmatrix} 77.17 & 73.34 \\ 61.74 & 62.43 \\ 51.54 & 69.04 \\ 73.34 & 53.6 \\ 62.43 & 51.2 \\ 69.04 & 51.47 \\ 53.6 & 51.58 \end{bmatrix} \quad (40)$$

В первом столбце матрице лаговых векторов содержатся элементы временного ряда, во втором столбце содержатся элементы с временной задержкой τ равной 3. Вычислим матрицу расстояний от текущего элемента до других элементов на первом шаге:

$$\begin{bmatrix} 0 & 15.43 & 25.63 & 3.83 & 14.47 & 8.13 & 23.57 \\ 15.43 & 0 & 10.2 & 11.6 & 0.69 & 7.3 & 8.14 \\ 25.63 & 10.2 & 0 & 21.8 & 10.89 & 17.5 & 2.06 \\ 3.83 & 11.6 & 21.8 & 0 & 10.91 & 4.3 & 19.74 \\ 14.47 & 0.69 & 10.89 & 10.91 & 0 & 6.61 & 8.83 \\ 8.13 & 7.3 & 17.5 & 4.3 & 6.61 & 0 & 15.44 \\ 23.57 & 8.14 & 2.06 & 19.74 & 8.83 & 15.44 & 0 \end{bmatrix} \quad (41)$$

Для вычисления расстояний между элементами используется матричная норма Фробениуса, формула которой приведена в пункте 1.3.4. Далее для каждого элемента необходимо отобрать ближайшего соседа –

элемент с наименьшим расстоянием. Для текущей матрицы расстояний индексы ближайших соседей будут иметь следующий вид (42):

$$[3 \ 4 \ 6 \ 0 \ 1 \ 3 \ 2] \quad (42)$$

Вычислим отношения расстояний между точкой и ее ближайшим соседом на следующем и текущем шагах. Расстояния между точками на текущем шаге можно получить из матрицы расстояний (41). Расстояния между точками на следующем шаге вычисляются по последнему столбцу расширенной матрица лаговых векторов (40).

Для первого элемента значение на следующем шаге равняется 73.34, ближайшим соседом является 4-ый элемент, значение которого на следующем шаге равняется 53.6. Отношение расстояний R_1 будет равняться (43):

$$\frac{|53.6 - 73.34|}{3.83} = 5.15 \quad (43)$$

Вектор отношений расстояний R_j для глубины погружения t равной 1 представлен ниже (44):

$$[5.15 \ 16.2 \ 8.48 \ 5.15 \ 16.2 \ 0.49 \ 8.48] \quad (44)$$

Среди найденных R_j подсчитывается количество ложных соседей P_m , для которых R_j больше R , где R – некоторый заданный порог. В данной работе в качестве значения порогового R используется число 10. В рассматриваемом примере для глубины погружения t равной 1 было найдено 2 ложных соседа.

Тогда процент FNN для данной размерности будет равен:

$$F_m = \frac{P_m}{N} = \frac{2}{7} \quad (45)$$

Алгоритм поиска оптимальной размерности m завершается, когда процент FNN будет равняться нулю, для m равной 1 это не так, поэтому глубина погружения увеличивается на 1 и алгоритм переходит на следующую итерацию. Для m равной 2 расширенная матрица $m + 1$ имеет вид:

$$\begin{bmatrix} 77.17 & 73.34 & 53.6 \\ 61.74 & 62.43 & 51.21 \\ 51.54 & 69.04 & 51.47 \\ 73.34 & 53.6 & 51.58 \end{bmatrix} \quad (46)$$

Матрица расстояний от текущего элемента до других элементов равняется:

$$\begin{bmatrix} 0 & 18.9 & 25.99 & 20.11 \\ 18.9 & 0 & 12.16 & 14.58 \\ 25.99 & 12.16 & 0 & 26.72 \\ 20.11 & 14.58 & 26.72 & 0 \end{bmatrix} \quad (47)$$

Индексы элементов ближайших соседей для каждого элемента будут соответственно равняться:

$$[1 \ 2 \ 1 \ 1] \quad (48)$$

Вектор отношений расстояний R_j для глубины погружения m равной 2:

$$[0.13 \ 0.02 \ 0.02 \ 0.03] \quad (49)$$

Тогда процент FNN для данной размерности будет равен:

$$F_m = \frac{0}{4} = 0 \quad (50)$$

Таким образом, для исходного временного ряда (39) и временной задержке τ равной 3 оптимальным значением глубины погружения m будет 2.

Для временного ряда скорректированных цен закрытия акций компании Microsoft оптимальная величина глубины погружения в лаговое пространство m также равняется двум. Таким образом, для получения прогнозируемой цены акции на следующий день необходимо знать цену акции на текущий день и цену акции 21 дня назад. Исходный временной ряд, погруженный в лаговое пространство представлен на рисунке 31.

	$x(i)$	$x(i - \tau)$	y
2016-03-04	51.285620	52.470350	51.261435
2016-03-07	51.261435	52.384634	51.240180
2016-03-08	51.240180	52.301847	51.221855
2016-03-09	51.221855	52.221991	51.206461
2016-03-10	51.206461	52.145065	51.193996
...
2017-12-22	84.847178	81.990742	84.985351
2017-12-26	84.985351	82.124806	85.123720
2017-12-27	85.123720	82.259065	85.262284
2017-12-28	85.262284	82.393521	85.401044
2017-12-29	85.401044	82.528172	85.540000

460 rows × 3 columns

Рисунок 31 – Временной ряд, погруженный в лаговое пространство

Полученная матрица лаговых векторов будет использоваться далее для обучения нейронной сети прогнозированию поведения цен акций компании Microsoft. Полученные параметры временной задержки τ и размерности лагового пространства m для реконструкции аттрактора динамической системы уникальны для прогнозируемого временного ряда

2.4 Подбор параметров и обучение модели

Перед работой с нейронной сетью необходимо выполнить дополнительную предварительную обработку данных для приведения их к требуемому формату. В зависимости от распределения данных выбирается тип масштабирования. Так как исходные данные имеют нормальное распределение, будет выполняться процедура стандартизации, т.е. приведение наборов значений x к среднему значению \bar{X} равному нулю и стандартному отклонению σ_x равному единице. На рисунке 32 приведена таблица стандартизированных наборов значений обучающей выборки.

	$x(i)$	$x(i - \tau)$	y
2016-03-04	-1.352187	-1.157535	-1.358143
2016-03-07	-1.354670	-1.166971	-1.360319
2016-03-08	-1.356852	-1.176085	-1.362194
2016-03-09	-1.358733	-1.184876	-1.363770
2016-03-10	-1.360314	-1.193345	-1.365046
...
2017-12-22	2.093452	2.092274	2.093563
2017-12-26	2.107638	2.107033	2.107725
2017-12-27	2.121844	2.121813	2.121907
2017-12-28	2.136070	2.136615	2.136110
2017-12-29	2.150316	2.151438	2.150332

460 rows × 3 columns

Рисунок 32 – Таблица стандартизированных значений обучающей выборки

Перед обучением нейронной сети также необходимо разбить исходную выборку на обучающую и тестовую (валидационную) (пункт 1.4.4). На практике процесс разбиения выборки производится случайным образом, для

случая с прогнозированием данный метод не подходит, так как необходимо, чтобы нейронная сеть не знала данных, которые ей предстоит спрогнозировать. Для этого выборка явно разделяется на две части в некотором соотношении. В данной работе разделение происходит в соотношении 4 к 1. На рисунке 33 представлена схема, демонстрирующая распределение выборки цен акций компании Microsoft за два года на обучающую и тестовую выборки в данном соотношении.



Рисунок 33 – Разделение выборки цен акций компании Microsoft на обучающую и тестовую в соотношении 4 к 1

Таким образом, в тестовую выборку попадают цен акций за период 04.03.2016 – 17.08.2017, а в тестовую (валидационную) попадают цены акций за период 18.08.2017 – 29.12.2017. Прогнозирование будет выполняться на одну неделю 18.08.2017 – 28.08.2017.

В качестве нейронной сети для прогнозирования временных рядов будет использоваться нейронная сеть прямого распространения многослойный персептрон с сигмоидальной функцией гиперболического тангенса в качестве функции активации, обучаемая с помощью квазиньютоновского алгоритма Бройдена-Флетчера-Гольдфарба-Шанно с ограниченным использованием памяти, описанного в пункте 1.4.5.

Для выполнения прогнозирования необходимо подобрать количество и размер скрытых слоев многослойного персептрона. Подбор оптимальных параметров нейронной сети выполнялся экспериментальным методом с помощью сравнения результирующей среднеквадратической ошибки MSE

для разных наборов скрытых слоев. В эксперименте участвовали модели с одним, двумя и тремя скрытыми слоями и различным количеством нейронов на каждом слое (25, 50 или 100 нейронов). На рисунке 34 представлен график зависимости среднеквадратической ошибки прогноза многослойного персептрона от выбранного количества и размеров скрытых слоев.

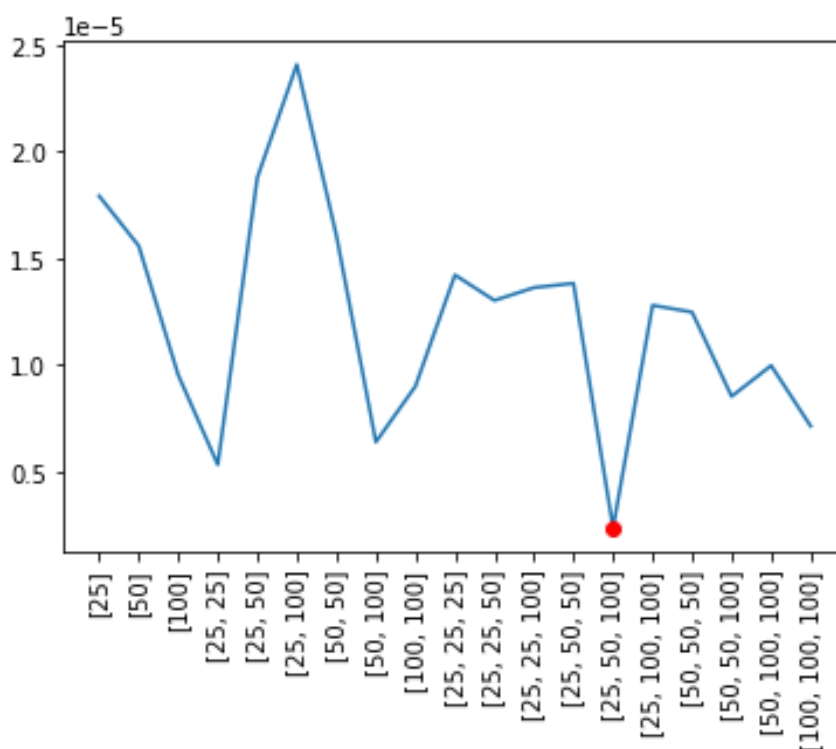


Рисунок 34 – График зависимости среднеквадратической ошибки прогноза от выбранного количества и размеров скрытых слоев

Как можно судить по графику, для данного примера наименьшая величина среднеквадратической ошибки получена для многослойного персептрона с тремя скрытыми слоями по [25, 50, 100] нейронов на соответствующих слоях. Оптимальная архитектура скрытых слоев модели подбирается для каждой выборки отдельно для повышения качества выполняемого прогнозирования.

Выбор обученной модели, выдающей наиболее точный прогноз, осуществляется следующим образом:

1. Создается множество моделей, по 3 модели на каждую возможную архитектуру скрытых слоев;
2. Последнее значение из обучающей выборки выделяется для прогноза и удаляется из обучающей выборки;
3. Каждая модель обучается на обучающей выборке и результат прогноза на отмеченный день сравнивается с точным значением с помощью метрики среднеквадратической ошибки MSE;
4. Среди множества моделей отбирается модель с наименьшим показателем ошибки. Данная модель будет использоваться далее для выполнения прогноза.

Для исходного временного ряда цен акций компании Microsoft экспериментальным путем была подобрана модель с тремя скрытыми слоями по [25, 50, 100] нейронов на слоях соответственно, которая имеет наименьший показатель среднеквадратической ошибки при прогнозировании на один день. Данная модель будет использоваться для выполнения прогноза на неделю.

2.5 Выполнение прогнозирования и оценка результатов

После отбора наилучшей обученной модели выполняется непосредственное прогнозирование. Для оценки полученного прогноза используется коэффициент детерминации R^2 , описанный в пункте 1.4.6. На рисунке 35 приведен график спрогнозированных значений временного ряда на 7 дней (18.08.2017 – 28.08.2017), полученных с помощью подобранной наилучшей модели с тремя скрытыми слоями по [25, 50, 100] нейронов.

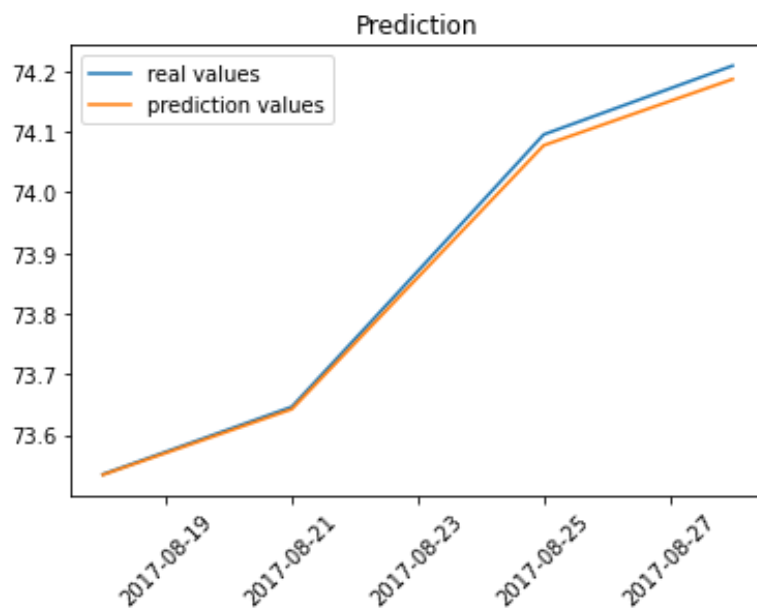


Рисунок 35 – График сравнения реальных и спрогнозированных значений временного ряда скорректированных цен закрытия акций компании Microsoft

Используемая модель при прогнозировании на один день имела показатель среднеквадратической ошибки MSE равный 0.0000015, что является наилучшим показателем среди других модель, участвующих в эксперименте. При прогнозировании на неделю данная модель показала среднеквадратическую ошибку 0.0000019 и коэффициент детерминации 0.9969, что демонстрирует высокую точность прогноза. По полученному графику сравнения реальных и спрогнозированных значений также можно судить о высоком качестве выполненного прогноза.

2.6 Выводы по разделу

Таким образом, в данном разделе была описана программная реализация алгоритма прогнозирования финансовых временных рядов с помощью нейронной сети многослойного персептрона. Для выполнения программирования методов, описанных в теоретической части работы, был

использован язык программирования Python и его основные библиотеки, такие как scikit-learn, NumPy, SciPy, pandas, matplotlib.

Для демонстрации работоспособности разработанной программы выполнялось прогнозирование временного ряда скорректированных цен закрытия акций компаний Microsoft за период 01.01.2016-01.01.2018 с шагом в один день. Информация о ценах акций компаний за выбранный промежуток времени была получена с помощью открытого источника Quandl. По каждому этапу работы программы выводились результаты работы с их описанием.

При описании частей программы для прогнозирования использовались иллюстрации блок-схем алгоритмов и приведены результаты их работы для временного ряда из примера. Там, где текстового описания и блок-схемы было недостаточно, например, при объяснении реализации алгоритма ложных ближайших соседей FNN, был приведен пример работы алгоритма с подробным разбором выполняемых действий по шагам.

Перед обучением нейронной сети исходная выборка явно разделялась на две части (обучающую и тестовую выборки) в соотношении 4 к 1. Подбор оптимальных параметров нейронной сети для выполнения прогнозирования выполнялся экспериментальным методом с помощью сравнения результирующей среднеквадратической ошибки MSE для разных наборов скрытых слоев. В эксперименте участвовали модели с одним, двумя и тремя скрытыми слоями и различным количеством нейронов на каждом слое (25, 50 или 100 нейронов). Для исходного временного ряда цен акций компании Microsoft была подобрана модель с тремя скрытыми слоями по [25, 50, 100] нейронов на слоях соответственно.

При прогнозировании временного ряда цен закрытия акций компании Microsoft на неделю данная модель показала среднеквадратическую ошибку 0.0000019 и коэффициент детерминации 0.9969, что демонстрирует высокую точность выполненного прогноза.

3 Тестирование программы и анализ результатов

3.1 Тестирование разработанной программы для прогнозирования временных рядов акций нескольких компаний

Для тестирования было выполнено прогнозирование поведения временных рядов цен закрытия акций компаний Microsoft, Apple, Intel, Disney, Visa и Coca-Cola на 7 дней (рисунок 36).

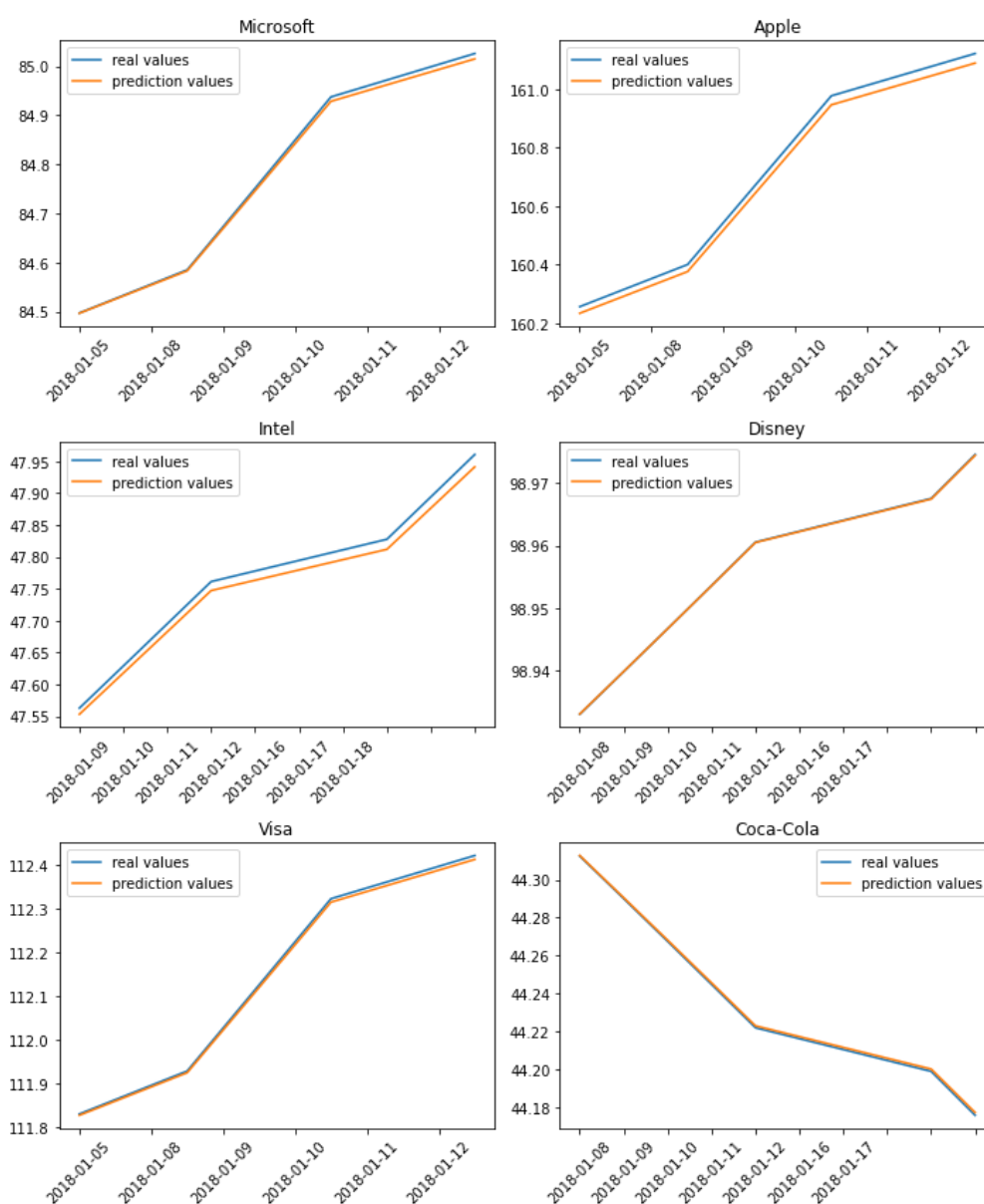


Рисунок 36 – Графики прогнозов временных рядов на 7 дней

Для обучения моделей использовались данные о ценах акций компаний за предыдущие два года. На рисунке 37 приведена блок-схема алгоритма выполнения прогноза цен акций компаний из списка на заданное количество дней.



Рисунок 37 – Блок-схема алгоритма выполнения прогноза

Список компаний предварительно формируется в программе и в цикле осуществляется проход по данному списку с выполнением описанного на предыдущей странице (рисунок 37) алгоритма. Получение временного ряда цен акций компаний осуществляется через обращение к сервису Quandl. Для выполнения погружения в лаговое пространство предварительно выполняется определение оптимальных значений временной задержки τ с помощью метода средней взаимной информации и размерности лагового пространства m с помощью метода ложных ближайших соседей FNN.

3.2 Анализ результатов прогнозирования

После выполнения прогноза для каждой компании из списка вычисляются показатели качества выполненного прогноза, описанные в пункте 1.4.6: среднеквадратическая ошибка MSE и коэффициент детерминации R^2 . Оценки результатов прогнозирования приведены в таблице на рисунке 38.

company	R2	MSE
Microsoft	0.998629	4.256374e-05
Apple	0.990815	7.645894e-04
Intel	0.987779	2.151932e-04
Disney	0.999975	4.727127e-09
Visa	0.999020	3.831100e-05
Coca-Cola	0.999586	8.550305e-07

Рисунок 38 – Оценки результатов прогноза

3.3 Выводы по разделу

Таким образом, было выполнено тестирование работоспособности программы для прогнозирования цен акций разных компаний на неделю и выполнена оценка результатов прогнозирования. В эксперименте участвовали временные ряды скорректированных цен закрытия акций компаний Microsoft, Apple, Intel, Disney, Visa и Coca-Cola за период 2016-2018 года с шагом в один день.

В тексте раздела приведено описание полного алгоритма разработанной программы по этапам в виде блок-схемы и последующего текстового описания.

Результаты выполненных тестовых прогнозов также приведены в тексте данного раздела. В качестве результатов приводятся графики сравнения реальных и спрогнозированных значений временных рядов и таблица оценок результатов: коэффициенты детерминации и оценки среднеквадратической ошибки для каждого временного ряда. Анализируя полученные графики и оценки прогнозов можно сделать вывод об успешности применения нейронной сети многослойного персептрона для задачи прогнозирования.

Заключение

В ходе выполнения выпускной квалификационной работы было выполнено исследование задачи прогнозирования финансовых временных рядов с помощью использования искусственной нейронной сети. В процессе выполнения работы была проанализирована необходимая литература и сформированы знания о предметной части, после чего выполнена разработка алгоритма прогнозирования и произведена его программная реализация.

В ходе работы был выполнен ряд поставленных в начале работы задач:

1. Исследована предметная область;
2. Выбран метод фильтрации временного ряда EMD, а также выполнена реконструкция аттрактора динамической системы, поведение которой подлежит прогнозированию, с помощью погружения временного ряда в лаговое пространство с параметрами временной задержки лага τ и размерности лагового пространства m (глубины погружения);
3. Выбрана модель нейронной сети – многослойный Персептрон, параметры которой подбираются для каждого временного ряда индивидуально с помощью сравнения нескольких тестовых моделей в прогнозировании на один день;
4. Выполнено тестирование разработанной программы на нескольких временных рядах с выводом результатов полученных прогнозов;
5. Сформирован вывод о полученных результатах.

Для формирования мнения о качестве прогноза, получаемого с помощью нейронной сети, было выполнено прогнозирования цен акций компаний Microsoft, Apple, Intel, Disney, Visa и Coca-Cola на 7 дней. Точность прогноза на неделю, полученного с помощью разработанной программы, составила 99 процентов, что позволяет сделать вывод об успешности применения нейронной сети многослойного персептрона для задачи прогнозирования.

Список используемой литературы

1. Аггарвал, Ч. Нейронные сети и глубокое обучение : учебный курс. – М. : Диалектика, 2020. – 744 с. : ил. – ISBN 978-5-907203-01-3.
2. Анатольев С.А. Задачи и решения по эконометрике. М. : Российская экономическая школа, 2005 г. – 164 с.
3. Андерсон Т. Статистический анализ временных рядов: Пер. с англ. – М. : Мир, 1976. 757 с.
4. Анищенко В.С., Астахов В.В., Вадивасова Т.Е. и др. Нелинейные эффекты в хаотических и стохастических системах / Под ред. В.С. Анищенко. Пер. с англ. М.-Ижевск : Институт компьютерных исследований, 2003. 529 с.
5. Безручко Б.П., Смирнов Д.А. Математическое моделирование и хаотические временные ряды. Саратов: ГосУНЦ «Колледж», 2005. 320 с. ISBN 5-94409- 045-6.
6. Верещагин Н.К., Щепин Е.В. Информация, кодирование и предсказание. – М. : ФМОП, МЦНМО, 2012. – 238 с.
7. Каток А.Б., Хассельблат Б. Введение в современную теорию динамических систем / Пер. с англ. М. : Факториал УРСС, 1999. 767 с.
8. Лоскутов А.Ю., Михайло А.С. Основы теории сложных систем. – М.-Ижевск : Институт компьютерных исследований, 2007. – 620 с.
9. Мишулина О.А. Статистический анализ и обработка временных рядов. – М.: МИФИ, 2004. – С. 180. – ISBN 5-7262-0536-7.
10. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. : Пер. с англ. – М. : Вильямс, 2017. – 480 с. – ISBN 978-5-9908910-8-1.
11. Николенко С., Кадурын А., Архангельская Е. Глубокое обучение. Погружение в мир нейронных сетей. – СПб. : Питер, 2018. – 480 с. : ил. – ISBN 978-5-496-02536-2.

12. Рашид, Т. Создаем нейронную сеть. : Пер. с англ. – СПб. : ООО “Альфа-книга”, 2017. – 272 с. : ил. – Парал. тит. англ. ISBN 978-5-9909445-7-2.
13. Введение в машинное обучение [Электронный ресурс]: Онлайн курс. / Образовательный сайт Coursera. URL: <https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie> (дата обращения: 11.01.2021).
14. Предсказание финансовых временных рядов [Электронный ресурс]: Лекция. / Национальный открытый университет ИНТУИТ. URL: <https://intuit.ru/studies/courses/2255/139/lecture/20571> (дата обращения: 15.12.2020).
15. Тренды и классификации [Электронный ресурс]: Онлайн курс по специализации «Анализ данных». / Образовательный сайт Coursera. URL: <https://www.coursera.org/learn/trendy-klassifikatsii> (дата обращения: 09.12.2020).
16. Avriel M. Nonlinear Programming: Analysis and Methods. – Dover Publishing, 2003. – 544 p. – ISBN 978-0-486-43227-4.
17. Diebold F.X. Elements of Forecasting. – 4th ed. – Thomson South-Western, 2007. – 748 p.
18. Diebold F.X., Mariano R.S. Comparing Predictive Accuracy // Journal of Business and Economic Statistics, vol. 13. – 1995. – pp. 253-265.
19. Gorodetski A., Ilyashenko Yu. Minimal and strange attractors, International Journal of Bifurcation and Chaos, vol. 6, no. 6. – 1996. – pp. 1177-1183.
20. Haykin Simon S. Neural Networks: A Comprehensive Foundation. – Prentice Hall, 1999. – 842 p. – ISBN 978-0-13-273350-2.
21. Huang, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. // Proc. R. Soc. Lond. A. – 1998. – Т. 454. – С. 903-995.

22. Nocedal J., Stephen J. Numerical Optimization. – 2nd edition. – USA: Springer, 2006. – 683 p. – ISBN 978-0-387-30303-1.
23. Rhodes C., Morari M. The false nearest neighbors' algorithm: An overview. Computers & Chemical Engineering. – 1991 – S1149-S1154. doi:10.1016/S0098-1354(97)87657-0.
24. Shumway R. H., Stoffer D. S. Time Series Analysis and Its Applications: With R Examples // Springer 3rd ed. 2011 edition. – 2011. – 596 p. – ISBN 978-1441978646.